SIEMENS

SIMATIC

Programmieren mit STEP 7 V5.1

Handbuch

Dieses Handbuch ist Bestandteil des Dokumentationspaketes mit der Bestellnummer:

6ES7 810-4CA05-8AA0

Ausgabe 08/2000
A5E00069872-03

Wichtige Hinweise, Inhaltsverzeichnis
Produkteinführung 1
Installieren und Autorisierung 2
Konzeption der Automatisierungslösung 3
Grundlagen zum Entwerfen einer Programmstruktur 4
Starten und Bedienen 5
Einrichten und Bearbeiten des Projekts 6
Festlegen von Symbolen 7
Anlegen von Bausteinen und Bibliotheken 8
Erstellen von Codebausteinen 9
Erstellen von Datenbausteinen 10
Erstellen von AWL-Quellen 11
Anzeigen von Referenzdaten 12
Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft 13
Projektieren von Meldungen 14
Bedienen und Beobachten von Variablen 15
Aufbau der Online-Verbindung und CPU-Einstellungen 16
Laden 17
Testen mit der Variablentabelle 18
Testen mit Programmstatus 19
Testen mit dem Simulationsprogramm S7-PLCSIM (Optionspaket) 20
Diagnose 21
Drucken und Archivieren 22
Mehrere Personen bearbeiten ein Projekt 23
Arbeiten mit M7-Automatisierungssystemen 24
Tips und Tricks 25
Anhang A
Index

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:

Gefahr
bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten werden, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden

Warnung
bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Vorsicht
bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis
ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von qualifiziertem Personal vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:

Warnung
Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind eingetragene Marken der Siemens AG.

Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright © Siemens AG 2000 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

Siemens AG
Bereich Automatisierungs- und Antriebstechnik (A&D)
Geschäftsgebiet Industrie-Automatisierungssysteme (AS)
Postfach 4848, D- 90327 Nürnberg

Haftungsausschluß

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard-und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 2000
Technische Änderungen bleiben vorbeh.

Excellence in Automation & Drives: Siemens

Wichtige Hinweise

Zweck des Handbuchs

Dieses Handbuch gibt Ihnen einen vollständigen Überblick über das Programmieren mit **STEP 7**. Es unterstützt Sie bei der Installation und Inbetriebnahme der Software. Die Vorgehensweise für die Programmerstellung und die Bestandteile von Anwenderprogrammen werden erläutert.

Das Handbuch richtet sich an Personen, die für die Realisierung von Steuerungsaufgaben mit STEP 7 auf Basis der SIMATIC S7 Automatisierungssysteme tätig sind.

Wir empfehlen Ihnen, sich mit den Beispielen aus dem Handbuch Getting Started „Erste Schritte und Übungen mit STEP 7“ vertraut zu machen. Sie bieten einen leichten Einstieg in die Thematik „Programmieren mit STEP 7“.

Erforderliche Grundkenntnisse

Zum Verständnis des Handbuchs sind allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik erforderlich.

Außerdem werden Kenntnisse über die Verwendung von Computern oder PC-ähnlichen Arbeitsmitteln (z. B. Programmiergeräten) unter dem Betriebssystem Windows 95/98/2000 bzw. NT vorausgesetzt.

Gültigkeitsbereich des Handbuchs

Das Handbuch ist gültig für das Softwarepaket STEP 7 V5.1.

Dokumentationspakete zu STEP 7

Dieses Handbuch ist Bestandteil des Dokumentationspaketes „STEP 7 Grundwissen“.

Die folgende Tabelle zeigt die Dokumentation zu STEP 7 im Überblick:

Handbücher	Zweck	Bestell-Nummer
STEP 7-Grundwissen mit • Erste Schritte und Übungen mit STEP 7 V5.1 • Programmieren mit STEP 7 V5.1 • Hardware konfigurieren und Verbindungen projektieren mit STEP 7 V5.1 • Von S5 nach S7, Umsteigerhandbuch	Das Grundwissen für technisches Personal, das das Vorgehen zur Realisierung von Steuerungsaufgaben mit STEP 7 und S7-300/400 beschreibt.	6ES7810-4CA05-8AA0
STEP 7-Referenzwissen mit • Handbücher KOP/FUP/AWL für S7-300/400 • Standard- und Systemfunktionen für S7-300/400	Das Referenzwissen zum Nachschlagen, das die Programmiersprachen KOP, FUP und AWL sowie Standard- und Systemfunktionen ergänzend zum STEP 7-Grundwissen beschreibt.	6ES7810-4CA05-8AR0

Online-Hilfen	Zweck	Bestell-Nummer
Hilfe zu STEP 7	Das Grundwissen zum Programmieren und Hardware konfigurieren mit STEP 7 als Online-Hilfe	Bestandteil des Softwarepaketes STEP 7
Referenzhilfen zu AWL/KOP/FUP Referenzhilfe zu SFBs/SFCs Referenzhilfe zu Organisationsbausteinen	Kontextsensitives Referenzwissen	Bestandteil des Softwarepaketes STEP 7

Online-Hilfe

Ergänzend zum Handbuch erhalten Sie bei der Nutzung der Software detaillierte Unterstützung durch die in die Software integrierte Online-Hilfe.

Das Hilfesystem ist über mehrere Schnittstellen in die Software integriert:

- Im Menü **Hilfe** stehen mehrere Menübefehle zur Verfügung: **Hilfethemen** öffnet das Inhaltsverzeichnis der Hilfe zu STEP 7.
- **Hilfe benutzen** gibt detaillierte Anweisungen zum Umgang mit der Online-Hilfe.
- Die kontext-sensitive Hilfe bietet Informationen zum aktuellen Kontext, z. B. zu einem geöffneten Dialogfeld oder zu einem aktiven Fenster. Sie läßt sich über die Schaltfläche "Hilfe" oder über die Taste F1 aufrufen.
- Eine weitere Form kontext-sensitiver Hilfe bietet die Statuszeile. Zu jedem Menübefehl wird hier eine kurze Erklärung angezeigt, sobald sich der Mauszeiger auf dem Menübefehl befindet.
- Auch zu den Symbolen in der Funktionsleiste wird eine kurze Erläuterung eingeblendet, wenn der Mauszeiger kurze Zeit über den Symbolen verweilt.

Wenn Sie Informationen der Online-Hilfe lieber in gedruckter Form lesen möchten, können Sie einzelne Hilfethemen, Bücher oder die gesamte Hilfe auch ausdrucken.

Dieses Handbuch ist ein Auszug der HTML basierenden Hilfe zu STEP 7. Detaillierte Handlungsanweisungen finden Sie in der Hilfe zu STEP 7. Aufgrund der nahezu identischen Gliederungsstruktur von Handbuch und Online-Hilfe können Sie bequem zwischen Handbuch und Online-Hilfe wechseln.

Rückmeldung zur Dokumentation

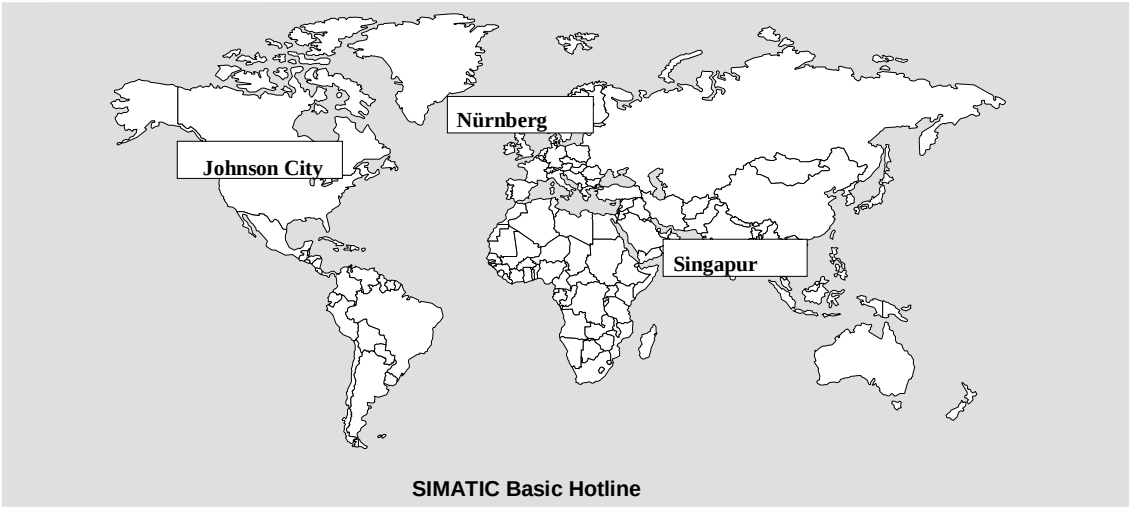
Um Ihnen und zukünftigen Anwendern eine optimale Dokumentation anbieten zu können, bitten wir Sie, uns hierbei zu unterstützen. Bei Anmerkungen zum vorliegenden *Handbuch* oder zur *Online-Hilfe* füllen Sie bitte den Fragebogen am Ende des Handbuchs aus, und schicken Sie ihn an die dort angegebene Adresse. Bitte tragen Sie dort auch Ihre persönliche Bewertung ein.

SIMATIC Trainingscenter

Um Ihnen den Einstieg in das Automatisierungssystem SIMATIC S7 zu erleichtern, bieten wir entsprechende Kurse an. Wenden Sie sich bitte an Ihr regionales Trainingscenter oder an das zentrale Trainingscenter in D 90327 Nürnberg.
Telefon: +49 (911) 895-3200.

SIMATIC Customer Support Hotline

Weltweit erreichbar zu jeder Tageszeit:



SIMATIC Basic Hotline

Weltweit (Nürnberg)
Technical Support
(FreeContact)

Ortszeit:	Mo.-Fr. 7:00 bis 17:00
Telefon:	+49 (180) 5050 222
Fax:	+49 (180) 5050 223
E-Mail:	techsupport@ ad.siemens.de
GMT:	+1:00

Weltweit (Nürnberg)
Technical Support
(kostenpflichtig, nur mit SIMATIC Card)

Ortszeit:	Mo.-Fr. 0:00 bis 24:00
Telefon:	+49 (911) 895-7777
Fax:	+49 (911) 895-7001
GMT:	+01:00

Europa / Afrika (Nürnberg)
Autorisierung

Ortszeit:	Mo.-Fr. 7:00 bis 17:00
Telefon:	+49 (911) 895-7200
Fax:	+49 (911) 895-7201
E-Mail:	authorization@ nbgm.siemens.de
GMT:	+1:00

Amerika (Johnson City)
Technical Support und Autorisierung

Ortszeit:	Mo.-Fr. 8:00 bis 19:00
Telefon:	+1 423 461-2522
Fax:	+1 423 461-2289
E-Mail:	simatic.hotline@ sea.siemens.com
GMT:	-5:00

Asien / Autralien (Singapur)
Technical Support und Autorisierung

Ortszeit:	Mo.-Fr. 8:30 bis 17:30
Telefon:	+65 740-7000
Fax:	+65 740-7001
E-Mail:	simatic.hotline@ sae.siemens.com.sg
GMT:	+8:00

Die Sprachen an den SIMATIC Hotlines sind generell Deutsch und Englisch, bei der Autorisierungshotline wird zusätzlich Französisch, italienisch und Spanisch gesprochen.

SIMATIC Customer Support Online-Dienste

Das SIMATIC Customer Support bietet Ihnen über die Online-Dienste umfangreiche zusätzliche Informationen zu den SIMATIC-Produkten:

- Allgemeine aktuelle Informationen erhalten Sie
  - im **Internet** unter http://www.ad.siemens.de/simatic
- Aktuelle Produkt-Informationen und Downloads, die beim Einsatz nützlich sein können:
  - im **Internet** unter http://www.ad.siemens.de/simatic-cs
  - über das **Bulletin Board System** (BBS) in Nürnberg (*SIMATIC Customer Support Mailbox)* unter der Nummer +49 (911) 895-7100.

    Verwenden Sie zur Anwahl der Mailbox ein Modem mit bis zu V.34 (28,8 kBaud), dessen Parameter Sie wie folgt einstellen: 8, N, 1, ANSI, oder wählen Sie sich per ISDN (x.75, 64 kBit) ein.
- Ihren Ansprechpartner für Automation & Drives vor Ort finden Sie über unsere Ansprechpartner-Datenbank:
  - im **Internet** unter http://www3.ad.siemens.de/partner/search.asp

Inhaltsverzeichnis

1 Produkteinführung 1-1

1.1 Leitfaden durch STEP 7 ... 1-1
1.2 Das STEP 7 - Basispaket ... 1-6
1.3 Was ist neu in STEP 7 Version 5.1 ... 1-10
1.4 Erweiterungsmöglichkeiten des STEP 7 - Basispakets ... 1-14
1.4.1 Erweiterungsmöglichkeiten des STEP 7 - Basispakets ... 1-14
1.4.2 Engineering Tools ... 1-15
1.4.3 Runtime Software ... 1-17
1.4.4 Human Machine Interface ... 1-18

2 Installieren und Autorisierung 2-1

2.1 Autorisierung ... 2-1
2.1.1 Autorisierung ... 2-1
2.1.2 Installieren und Deinstallieren der Autorisierung ... 2-1
2.1.3 Regeln für den Umgang mit Autorisierungen ... 2-5
2.2 Installieren von STEP 7 ... 2-8
2.2.1 Installieren von STEP 7 ... 2-8
2.2.2 Vorgehen beim Installieren ... 2-9
2.2.3 Einstellen der PG/PC-Schnittstelle ... 2-12
2.3 Deinstallieren von STEP 7 ... 2-14
2.3.1 Deinstallieren von STEP 7 ... 2-14

3 Konzeption der Automatisierungslösung 3-1

3.1 Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung ... 3-1
3.2 Zerlegen des Prozesses in Aufgaben und Bereiche ... 3-2
3.3 Beschreiben der einzelnen Funktionsbereiche ... 3-4
3.4 Auflisten von Ein-, Aus- und Durchgängen ... 3-6
3.5 Erstellen eines Eingangs-, Ausgangs-Diagramms für die Motoren ... 3-6
3.6 Erstellen eines Eingangs-/ Ausgangs-Diagramms für die Ventile ... 3-7
3.7 Festlegen der Sicherheitsanforderungen ... 3-7
3.8 Beschreiben der notwendigen Anzeige- und Bedienelemente ... 3-9
3.9 Erstellen des Konfigurationsplans ... 3-10

4 Grundlagen zum Entwerfen einer Programmstruktur 4-1

4.1 Programme in einer CPU ... 4-1
4.2 Bausteine im Anwenderprogramm ... 4-2
4.2.1 Bausteine im Anwenderprogramm ... 4-2
4.2.2 Organisationsbausteine und Programmstruktur ... 4-3
4.2.3 Aufrufhierarchie im Anwenderprogramm ... 4-9
4.2.4 Bausteinarten und zyklische Programmbearbeitung ... 4-11
4.2.4.1 Organisationsbaustein für zyklische Programmbearbeitung (OB 1) ... 4-11
4.2.4.2 Funktionen (FC) ... 4-16
4.2.4.3 Funktionsbausteine (FB) ... 4-17
4.2.4.4 Instanz-Datenbausteine ... 4-20
4.2.4.5 Globale Datenbausteine (DB) ... 4-22
4.2.4.6 Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC) ... 4-23
4.2.5 Organisationsbausteine für alarmgesteuerte Programmbearbeitung ... 4-24

4.2.5.1 Organisationsbausteine für alarmgesteuerte Programmbearbeitung 4-24
4.2.5.2 Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17) 4-25
4.2.5.3 Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23) 4-27
4.2.5.4 Organisationsbausteine für den Weckalarm (OB 30 bis OB 38) 4-27
4.2.5.5 Organisationsbausteine für den Prozeßalarm (OB 40 bis OB 47) 4-29
4.2.5.6 Organisationsbausteine für den Anlauf (OB 100 / OB 101 / OB 102) 4-30
4.2.5.7 Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90) 4-32
4.2.5.8 Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122) 4-33

5 Starten und Bedienen 5-1

5.1 Starten des SIMATIC Managers 5-1
5.1.1 Starten von STEP 7 5-1
5.1.2 Starten von STEP 7 mit vorgegebenen Startparametern 5-2
5.1.3 Aufrufen der Hilfefunktionen 5-4
5.2 Objekte und Objekthierarchie 5-5
5.2.1 Objekte und Objekt-Hierarchie 5-5
5.2.2 Objekt Projekt 5-6
5.2.3 Objekt Bibliothek 5-7
5.2.4 Objekt Station 5-8
5.2.5 Objekt Programmierbare Baugruppe 5-9
5.2.6 Objekt S7/M7-Programm 5-11
5.2.7 Objekt Baustein-Ordner 5-12
5.2.8 Objekt Quellen-Ordner 5-15
5.2.9 S7/M7-Programm ohne Station und CPU 5-16
5.3 Benutzungsoberfläche und Bedienung 5-18
5.3.1 Bedienphilosophie 5-18
5.3.2 Aufbau des Fensters 5-19
5.3.3 Elemente in Dialogfeldern 5-19
5.3.4 Erzeugen und Manipulieren von Objekten 5-21
5.3.5 Auswahl von Objekten im Dialog 5-26
5.3.6 Sitzungsgedächtnis 5-28
5.3.7 Ändern der Fensteranordnung von Symboltabellen 5-28
5.3.8 Speichern und Wiederherstellen der Fensteranordnung 5-29
5.4 Bedienung über Tastureingabe 5-30
5.4.1 Bedienung über Tastatureingabe 5-30
5.4.2 Tastenkombinationen für Menübefehle 5-30
5.4.3 Tastenkombinationen für das Bewegen des Cursors 5-32
5.4.4 Tastenkombinationen für das Markieren von Texten 5-33
5.4.5 Tastenkombinationen für den Zugriff auf Online-Hilfe 5-33
5.4.6 Tastenkombinationen für das Umschalten zwischen Fenstertypen 5-34

6 Einrichten und Bearbeiten des Projekts 6-1

6.1 Projektstruktur 6-1
6.2 Einrichten eines Projekts 6-3
6.2.1 Anlegen eines Projekts 6-3
6.2.2 Einfügen von Stationen 6-4
6.2.3 Einfügen eines S7- / M7-Programms 6-6
6.3 Bearbeiten eines Projekts 6-8
6.3.1 Bearbeiten eines Projekts 6-8
6.3.2 Texte mehrsprachig verwalten 6-9

7 Festlegen von Symbolen 7-1

7.1 Absolute und symbolische Adressierung ... 7-1
7.2 Globale und lokale Symbole ... 7-3
7.3 Darstellung von globalen oder lokalen Symbolen ... 7-4
7.4 Einstellung des Operandenvorrangs ... 7-5
7.5 Symboltabelle für globale Symbole ... 7-6
7.5.1 Symboltabelle für globale Symbole ... 7-6
7.5.2 Struktur und Bestandteile der Symboltabelle ... 7-6
7.5.3 Zulässige Adressen und Datentypen in der Symboltabelle ... 7-8
7.5.4 Unvollständige und mehrdeutige Symbole in der Symboltabelle ... 7-9
7.6 Eingabemöglichkeiten von globalen Symbolen ... 7-10
7.6.1 Eingabemöglichkeiten von globalen Symbolen ... 7-10
7.6.2 Allgemeine Hinweise zur Eingabe von Symbolen ... 7-10
7.6.3 Eingeben einzelner globaler Symbole im Dialog ... 7-11
7.6.4 Eingeben mehrerer globaler Symbole in der Symboltabelle ... 7-12
7.6.5 Groß-/Kleinschreibung bei Symbolen ... 7-13
7.6.6 Exportieren und Importieren von Symboltabellen ... 7-15
7.6.7 Dateiformate für den Import/Export einer Symboltabelle ... 7-16

8 Anlegen von Bausteinen und Bibliotheken 8-1

8.1 Auswahl der Erstellmethode ... 8-1
8.2 Auswahl der Programmiersprache ... 8-2
8.2.1 Auswahl der Programmiersprache ... 8-2
8.2.2 Programmiersprache KOP (Kontaktplan) ... 8-4
8.2.3 Programmiersprache FUP (Funktionsplan) ... 8-4
8.2.4 Programmiersprache AWL (Anweisungsliste) ... 8-5
8.2.5 Programmiersprache S7-SCL ... 8-5
8.2.6 Programmiersprache S7-GRAPH (Ablaufsteuerung) ... 8-6
8.2.7 Programmiersprache S7-HiGraph (Zustandsgraph) ... 8-7
8.2.8 Programmiersprache S7-CFC ... 8-9
8.3 Anlegen von Bausteinen ... 8-10
8.3.1 Baustein-Ordner ... 8-10
8.3.2 Anwenderdefinierte Datentypen (UDT) ... 8-11
8.3.3 Bausteineigenschaften ... 8-12
8.3.4 Anzeige von Baustein-Längen ... 8-14
8.3.5 Umverdrahten ... 8-15
8.3.6 Attribute für Bausteine und Parameter ... 8-16
8.4 Arbeiten mit Bibliotheken ... 8-17
8.4.1 Arbeiten mit Bibliotheken ... 8-17
8.4.2 Hierarchischer Aufbau von Bibliotheken ... 8-19
8.4.3 Übersicht der Standardbibliotheken ... 8-19

9 Erstellen von Codebausteinen 9-1

9.1 Grundlagen zum Erstellen von Codebausteinen ... 9-1
9.1.1 Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen ... 9-1
9.1.2 Voreinstellungen für Programmeditor KOP/FUP/AWL ... 9-2
9.1.3 Zugriffsrechte auf Bausteine bzw. Quellen ... 9-3
9.1.4 Anweisungen aus dem Programmelemente-Katalog ... 9-4
9.2 Editieren der Variablendeklarationstabelle ... 9-5
9.2.1 Verwendung der Variablendeklaration in Codebausteinen ... 9-5
9.2.2 Zusammenspiel zwischen Variablendeklarationstabelle und Anweisungsteil ... 9-6
9.2.3 Aufbau der Variablendeklarationstabelle ... 9-6
9.2.4 Allgemeine Hinweise zu Variablendeklarationstabellen ... 9-8
9.3 Multiinstanzen in der Variablendeklarationstabelle ... 9-9
9.3.1 Verwendung von Multiinstanzen ... 9-9

9.3.2 Regeln für die Bildung von Multiinstanzen ... 9-10
9.3.3 Eingeben der Multiinstanz in der Variablendeklarationstabelle ... 9-10
9.4 Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren ... 9-11
9.4.1 Aufbau des Anweisungsteils ... 9-11
9.4.2 Vorgehensweise beim Eingeben von Anweisungen ... 9-12
9.4.3 Eingeben von globalen Symbolen in ein Programm ... 9-12
9.4.4 Titel und Kommentare zu Bausteinen und Netzwerken ... 9-13
9.4.5 Suchfunktion für Fehler im Anweisungsteil ... 9-14
9.5 Editieren von KOP-Anweisungen im Anweisungsteil ... 9-15
9.5.1 Einstellungen für Programmiersprache KOP ... 9-15
9.5.2 Regeln für die Eingabe von KOP-Anweisungen ... 9-16
9.5.3 Unzulässige Verschaltungen in KOP ... 9-18
9.6 Editieren von FUP-Anweisungen im Anweisungsteil ... 9-19
9.6.1 Einstellungen für die Programmiersprache FUP ... 9-19
9.6.2 -Regeln für die Eingabe von FUP-Anweisungen ... 9-20
9.7 Editieren von AWL-Anweisungen im Anweisungsteil ... 9-22
9.7.1 Einstellungen für die Programmiersprache AWL ... 9-22
9.7.2 Regeln für die Eingabe von AWL-Anweisungen ... 9-22
9.8 Aktualisieren von Bausteinaufrufen ... 9-23
9.8.1 Aktualisieren von Bausteinaufrufen ... 9-23
9.9 Speichern von Codebausteinen ... 9-24
9.9.1 Speichern von Codebausteinen ... 9-24
9.9.2 Korrigieren der Schnittstellen in einem FC, FB oder UDT ... 9-24
9.9.3 Vermeiden von Fehlern beim Aufrufen von Bausteinen ... 9-25

10 Erstellen von Datenbausteinen 10-1

10.1 Grundlagen zum Erstellen von Datenbausteinen ... 10-1
10.2 Deklarationssicht von Datenbausteinen ... 10-2
10.3 Datensicht von Datenbausteinen ... 10-3
10.4 Eingeben und Speichern in Datenbausteinen ... 10-4
10.4.1 Eingeben der Datenstruktur von globalen Datenbausteinen ... 10-4
10.4.2 Eingeben / Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs) ... 10-5
10.4.3 Eingeben der Struktur von anwenderdefinierten Datentypen (UDT) ... 10-6
10.4.4 Eingeben / Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT ... 10-7
10.4.5 Ändern von Datenwerten in der Datensicht ... 10-8
10.4.6 Rücksetzen von Datenwerten auf die Anfangswerte ... 10-8
10.4.7 Speichern von Datenbausteinen ... 10-9

11 Erstellen von AWL-Quellen 11-1

11.1 Grundlagen zum Programmieren in AWL-Quellen ... 11-1
11.2 Regeln zum Programmieren in AWL-Quellen ... 11-2
11.2.1 Regeln zur Eingabe von Anweisungen in AWL-Quellen ... 11-2
11.2.2 Regeln für Variablendeklaration in AWL-Quellen ... 11-3
11.2.3 Regeln zur Reihenfolge der Bausteine in AWL-Quellen ... 11-4
11.2.4 Regeln zum Festlegen von Systemattributen in AWL-Quellen ... 11-4
11.2.5 Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen ... 11-5
11.2.6 Zulässige Bausteineigenschaften je Bausteinart ... 11-7
11.3 Struktur von Bausteinen in AWL-Quellen ... 11-8
11.3.1 Struktur von Bausteinen in AWL-Quellen ... 11-8
11.3.2 Struktur von Codebausteinen in AWL-Quellen ... 11-8
11.3.3 Struktur von Datenbausteinen in AWL-Quellen ... 11-9
11.3.4 Struktur von anwenderdefinierten Datentypen in AWL-Quellen ... 11-9
11.4 Syntax und Formate für Bausteine in AWL-Quellen ... 11-10

11.4.1 Syntax und Formate für Bausteine in AWL-Quellen ... 11-10
11.4.2 Formattabelle von OBs ... 11-10
11.4.3 Formattabelle von FBs ... 11-11
11.4.4 Formattabelle von FCs ... 11-11
11.4.5 Formattabelle von DBs ... 11-12
11.5 Erstellen von AWL-Quellen ... 11-13
11.5.1 Anlegen von AWL-Quellen ... 11-13
11.5.2 Bearbeiten von S7-Quellen ... 11-13
11.5.3 Einfügen von Bausteinvorlagen in AWL-Quellen ... 11-14
11.5.4 Einfügen von externen Quellen ... 11-14
11.5.5 Generieren von AWL-Quellen aus Bausteinen ... 11-15
11.6 Speichern, Übersetzen von AWL-Quellen und Konsistenzprüfung ... 11-15
11.6.1 Speichern von AWL-Quellen ... 11-15
11.6.2 Prüfen der Konsistenz in AWL-Quellen ... 11-16
11.6.3 Fehlersuche in AWL-Quellen ... 11-16
11.6.4 Übersetzen von AWL-Quellen ... 11-16
11.7 Beispiele zu AWL-Quellen ... 11-18
11.7.1 Beispiele für Variablendeklarationen in AWL-Quellen ... 11-18
11.7.2 Beispiel für OBs in AWL-Quellen ... 11-19
11.7.3 Beispiel für FCs in AWL-Quellen ... 11-20
11.7.4 Beispiel für FBs in AWL-Quellen ... 11-22
11.7.5 Beispiele für DBs in AWL-Quellen ... 11-23
11.7.6 Beispiel für UDTs in AWL-Quellen ... 11-24

12 Anzeigen von Referenzdaten 12-1

12.1 Übersicht der möglichen Referenzdaten ... 12-1
12.1.1 Übersicht der möglichen Referenzdaten ... 12-1
12.1.2 Querverweisliste ... 12-2
12.1.3 Programmstruktur ... 12-4
12.1.4 Belegungsplan für Eingänge, Ausgänge und Merker (E/A/M) ... 12-6
12.1.5 Belegungsplan für Zeiten und Zähler (T/Z) ... 12-7
12.1.6 Nicht verwendete Symbole ... 12-8
12.1.7 Operanden ohne Symbol ... 12-9
12.1.8 Anzeige sprachabhängiger Informationen zu KOP, FUP, AWL ... 12-9
12.2 Arbeiten mit Referenzdaten ... 12-10
12.2.1 Möglichkeiten zum Anzeigen von Referenzdaten ... 12-10
12.2.2 Anzeigen von Listen in zusätzlichen Arbeitsfenstern ... 12-10
12.2.3 Erzeugen und Anzeigen von Referenzdaten ... 12-11
12.2.4 Schnelles Positionieren auf Verwendungsstellen im Programm ... 12-12
12.2.5 Beispiel zum Arbeiten mit Verwendungsstellen ... 12-13

13 Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft 13-1

13.1 Bausteinkonsistenz prüfen ... 13-1
13.2 Zeitstempel als Bausteineigenschaft und Zeitstempelkonflikte ... 13-3
13.3 Zeitstempel in Codebausteinen ... 13-4
13.4 Zeitstempel bei globalen Datenbausteinen ... 13-5
13.5 Zeitstempel bei Instanz-Datenbausteinen ... 13-6
13.6 Zeitstempel bei UDTs und von UDTs abgeleiteten DBs ... 13-7

14 Projektieren von Meldungen 14-1

14.1 Grundlagen des Meldekonzeptes ... 14-1
14.1.1 Grundlagen des Meldekonzeptes ... 14-1
14.1.2 Welche Meldeverfahren gibt es? ... 14-1
14.1.3 Auswahl des Meldeverfahrens ... 14-2
14.1.4 SIMATIC-Komponenten ... 14-4
14.1.5 Bestandteile einer Meldung ... 14-4
14.1.6 Vergabe von Meldenummern ... 14-5
14.2 Bausteinbezogene Meldungen zuordnen und bearbeiten ... 14-6
14.2.1 Bausteinbezogenen Meldungen zuordnen und bearbeiten ... 14-6
14.2.2 Welche Meldebausteine gibt es? ... 14-6
14.2.3 Formalparameter, Systemattribute und Meldebausteine ... 14-7
14.2.4 Meldungstyp und Meldungen ... 14-8
14.2.5 Bausteinbezogene Meldungen anlegen ... 14-9
14.2.6 PCS7-Meldungsprojektierung ... 14-12
14.3 Symbolbezogene Meldungen zuordnen und bearbeiten ... 14-14
14.3.1 Symbolbezogene Meldungen zuordnen und bearbeiten ... 14-14
14.4 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten ... 14-15
14.4.1 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten ... 14-15
14.5 Bedienerrelevante Texte übersetzen und bearbeiten ... 14-16
14.5.1 Bedienerrelevante Texte übersetzen und bearbeiten ... 14-16
14.5.2 Anwendertexte übersetzen und bearbeiten ... 14-16
14.5.3 Textbibliotheken übersetzen ... 14-17
14.6 Meldungsprojektierungsdaten zum Zielsystem transferieren ... 14-19
14.6.1 Projektierungsdaten zum Zielsystem transferieren ... 14-19
14.7 CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen ... 14-20
14.7.1 CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen ... 14-20
14.7.2 CPU-Meldungen konfigurieren ... 14-22
14.7.3 Anzeigen von gespeicherten CPU-Meldungen ... 14-22
14.8 'Melden von Systemfehlern' projektieren ... 14-23
14.8.1 'Melden von Systemfehlern' projektieren ... 14-23
14.8.2 Unterstützte Komponenten und Funktionsumfang ... 14-24
14.8.3 Einstellungen für das Melden von Systemfehlern ... 14-26
14.8.4 Bausteine für die Meldung von Systemfehlern generieren ... 14-27
14.8.5 Erzeugte Fehler-OBs ... 14-28
14.8.6 Erzeugter FB, DB ... 14-29

15 Bedienen und Beobachten von Variablen 15-1

15.1 Projektieren von bedien- und beobachtbaren Variablen ... 15-1
15.2 Projektieren von BuB-Attributen mit AWL, KOP und FUP ... 15-2
15.2.1 Projektieren von BuB-Attributen mit AWL, KOP und FUP ... 15-2
15.3 Projektieren von BuB-Attributen über die Symboltabelle ... 15-3
15.3.1 Projektieren von BuB-Attributen über die Symboltabelle ... 15-3
15.4 Abändern von BuB-Attributen mit CFC ... 15-4
15.4.1 Abändern von BuB-Attributen mit CFC ... 15-4
15.5 Transferieren der Projektierungsdaten zum BuB-Zielsystem ... 15-5
15.5.1 Transferieren der Projektierungsdaten zum BuB-Zielsystem ... 15-5

16 Aufbau der Online-Verbindung und CPU-Einstellungen 16-1

16.1 Aufbau von Online-Verbindungen ... 16-1
16.1.1 Aufbau von Online-Verbindungen ... 16-1
16.1.2 Online-Verbindung aufbauen über das Fenster "Erreichbare Teilnehmer" ... 16-1
16.1.3 Online-Verbindung aufbauen über das Online-Fenster des Projekts ... 16-2
16.1.4 Paßwortschutz für Zugriff auf Zielsysteme ... 16-3
16.1.5 Hinweis zur Aktualisierung des Fensterinhalts ... 16-4

16.2 Anzeigen und Ändern des Betriebszustands ... 16-4
16.2.1 Anzeigen und Ändern des Betriebszustands ... 16-4
16.3 Anzeigen und Einstellen von Uhrzeit und Datum ... 16-5
16.3.1 Anzeigen und Einstellen von Uhrzeit und Datum ... 16-5

17 Laden 17-1

17.1 Laden aus dem PG/PC in das Zielsystem ... 17-1
17.1.1 Voraussetzungen für das Laden ... 17-1
17.1.2 Unterschied zwischen Speichern und Laden von Bausteinen ... 17-2
17.1.3 Lade- und Arbeitsspeicher in der CPU ... 17-3
17.1.4 Lademöglichkeiten abhängig vom Ladespeicher ... 17-4
17.1.5 Nachladen von Bausteinen in das Zielsystem ... 17-5
17.1.6 Laden über EPROM-Memory Cards ... 17-6
17.1.7 Hardware-Konfiguration und Verbindungsprojektierung separat laden ... 17-7
17.1.7.1 Laden einer Konfiguration in ein Zielsystem ... 17-7
17.1.7.2 Erstmaliges Laden der Netzkonfiguration ... 17-8
17.1.7.3 Laden der Netzkonfiguration in ein Zielsystem ... 17-10
17.1.7.4 Laden von Änderungen der Netzkonfiguration ... 17-10
17.1.7.5 Laden der Globaldaten-Konfiguration ... 17-12
17.2 Laden aus dem Zielsystem in das PG ... 17-13
17.2.1 Laden aus dem Zielsystem in das PG ... 17-13
17.2.2 Station laden in PG ... 17-14
17.2.3 Zurückladen von Bausteinen aus S7-CPU ... 17-15
17.2.4 Bearbeiten von geladenen Bausteinen im PG/PC ... 17-15
17.2.5 Hardware-Konfiguration und Verbindungsprojektierung zurückladen ... 17-16
17.2.5.1 Zurückladen einer Konfiguration aus einer Station ... 17-16
17.2.5.2 Zurückladen einer Netzkonfiguration (Laden in PG) ... 17-17
17.2.6 Löschen im Zielsystem ... 17-19
17.2.6.1 Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU ... 17-19
17.2.6.2 Löschen von S7-Bausteinen auf dem Zielsystem ... 17-20
17.2.7 Komprimieren des Anwenderspeichers (RAM) ... 17-20
17.2.7.1 Entstehung von Lücken im Anwenderspeicher (RAM) ... 17-20
17.2.7.2 Komprimieren des Speicherinhalts einer S7-CPU ... 17-21

18 Testen mit der Variablentabelle 18-1

18.1 Einführung zum Testen mit Variablentabellen ... 18-1
18.2 Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablentabellen ... 18-2
18.3 Bearbeiten und Speichern von Variablentabellen ... 18-2
18.3.1 Erstellen und Öffnen einer Variablentabelle ... 18-2
18.3.2 Kopieren/Verschieben von Variablentabellen ... 18-3
18.3.3 Speichern einer Variablentabelle ... 18-3
18.4 Eingeben von Variablen in Variablentabellen ... 18-4
18.4.1 Einfügen von Operanden oder Symbolen in eine Variablentabelle ... 18-4
18.4.2 Einfügen von Steuerwerten ... 18-5
18.4.3 Obergrenzen für die Eingabe von Zeiten ... 18-6
18.4.4 Obergrenzen für die Eingabe von Zählern ... 18-7
18.4.5 Einfügen von Kommentarzeilen ... 18-7
18.5 Beispiele zur Eingabe in Variablentabellen ... 18-8
18.5.1 Beispiel für die Eingabe von Operanden in Variablentabellen ... 18-8
18.5.2 Beispiel für die Eingabe eines zusammenhängenden Operandenbereichs ... 18-9
18.5.3 Beispiele für die Eingabe von Steuer-/Forcewerten ... 18-9
18.6 Herstellen einer Verbindung zur CPU ... 18-12
18.6.1 Herstellen einer Verbindung zur CPU ... 18-12
18.7 Beobachten von Variablen ... 18-13

18.7.1 Einführung zum Beobachten von Variablen ... 18-13
18.7.2 Festlegen des Triggers zum Beobachten von Variablen ... 18-13
18.8 Steuern von Variablen ... 18-15
18.8.1 Einführung zum Steuern von Variablen ... 18-15
18.8.2 Festlegen des Triggers zum Steuern von Variablen ... 18-16
18.9 Forcen von Variablen ... 18-18
18.9.1 Einführung zum Forcen von Variablen ... 18-18
18.9.2 Sicherheitsvorkehrungen zum Forcen von Variablen ... 18-20
18.9.3 Unterschiede zwischen Forcen und Steuern von Variablen ... 18-21

19 Testen mit Programmstatus 19-1

19.1 Testen mit Programmstatus ... 19-1
19.2 Anzeigen im Programmstatus ... 19-3
19.3 Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte ... 19-5
19.4 Wissenswertes zum Betriebszustand HALT ... 19-7
19.5 Programmstatus von Datenbausteinen ... 19-8
19.6 Einstellen der Aufrufumgebung des Bausteins ... 19-9

20 Testen mit dem Simulationsprogramm S7-PLCSIM (Optionspaket) 20-1

20.1 Testen mit Simulationsprogramm (Optionspaket) ... 20-1

21 Diagnose 21-1

21.1 Diagnose der Hardware und Fehlersuche ... 21-1
21.2 Diagnosesymbole in der Online-Sicht ... 21-3
21.3 Hardware diagnostizieren: Schnellansicht ... 21-5
21.3.1 Aufruf der Schnellansicht ... 21-5
21.3.2 Auskunftsfunktionen der Schnellansicht ... 21-5
21.4 Hardware diagnostizieren: Diagnosesicht ... 21-6
21.4.1 Aufruf der Diagnosesicht von HW Konfig ... 21-6
21.4.2 Auskunftsfunktionen der Diagnosesicht ... 21-8
21.5 Baugruppenzustand ... 21-9
21.5.1.1 Aufrufmöglichkeiten des Baugruppenzustands ... 21-9
21.5.2 Auskunftsfunktionen des Baugruppenzustands ... 21-10
21.5.3 Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand ... 21-12
21.6 Diagnose im Betriebszustand STOP ... 21-14
21.6.1 Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache ... 21-14
21.6.2 Stack-Inhalte im Betriebszustand STOP ... 21-14
21.7 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern ... 21-16
21.7.1 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern ... 21-16
21.8 Übermittlung von Diagnoseinformationen ... 21-17
21.8.1 Übermittlung von Diagnoseinformationen ... 21-17
21.8.2 Systemzustandsliste SZL ... 21-18
21.8.3 Eigene Diagnosemeldungen senden ... 21-20
21.8.4 Diagnosefunktionen ... 21-21
21.9 Maßnahmen im Programm zur Störungsbehandlung ... 21-22
21.9.1 Maßnahmen im Programm zur Störungsbehandlung ... 21-22
21.9.2 Auswerten des Ausgangsparameter RET_VAL ... 21-23
21.9.3 Fehler-OBs als Reaktion auf die Erkennung eines Fehlers ... 21-24
21.9.4 Einfügen von Ersatzwerten bei Fehlererkennung ... 21-29
21.9.5 Peripherie-Redundanzfehler (OB 70) ... 21-31
21.9.6 CPU-Redundanzfehler (OB 72) ... 21-31
21.9.7 Kommunikations-Redundanzfehler (OB 73) ... 21-32
21.9.8 Zeitfehler (OB 80) ... 21-33
21.9.9 Stromversorgungsfehler (OB 81) ... 21-34

21.9.10 Diagnosealarm (OB 82) ... 21-35
21.9.11 Ziehen-/Stecken-Alarm (OB 83) ... 21-36
21.9.12 CPU-Hardwarefehler (OB 84) ... 21-37
21.9.13 Programmablauffehler (OB 85) ... 21-37
21.9.14 Baugruppenträgerausfall (OB 86) ... 21-38
21.9.15 Kommunikationsfehler (OB 87) ... 21-39
21.9.16 Programmierfehler (OB 121) ... 21-39
21.9.17 Peripheriezugriffsfehler (OB 122) ... 21-40

22 Drucken und Archivieren 22-1

22.1 Drucken einer Projektdokumentation ... 22-1
22.1.1 Drucken einer Projektdokumentation ... 22-1
22.1.2 Prinzipielle Vorgehensweise beim Drucken ... 22-2
22.1.3 Funktionen zum Drucken ... 22-2
22.1.4 Besonderheiten zum Drucken des Objektbaums ... 22-4
22.2 Archivieren von Projekten und Bibliotheken ... 22-4
22.2.1 Archivieren von Projekten und Bibliotheken ... 22-4
22.2.2 Anwendungsfälle für Speichern / Archivieren ... 22-5
22.2.3 Voraussetzungen für das Archivieren ... 22-6
22.2.4 Vorgehensweise beim Archivieren/Dearchivieren ... 22-6

23 Mehrere Personen bearbeiten ein Projekt 23-1

23.1 Mehrbenutzer-Konfiguration im Windows-Netzwerkverbund ... 23-1

24 Arbeiten mit M7-Automatisierungssystemen 24-1

24.1 Vorgehensweise für M7-Systeme ... 24-1
24.2 Optionssoftware für M7-Programmierung ... 24-3
24.3 Betriebssysteme für M7-300/400 ... 24-5

25 Tips und Tricks 25-1

25.1 Baugruppen tauschen in der Konfigurationstabelle ... 25-1
25.2 Projekte mit einer großen Anzahl vernetzter Stationen ... 25-1
25.3 Reorganisation ... 25-2
25.4 Testen mit der Variablentabelle ... 25-3
25.5 Virtueller Arbeitsspeicher ... 25-5

A Anhang A-1

A.1 Betriebszustände ... A-1
A.1.1 Betriebszustände und Übergänge ... A-1
A.1.2 Betriebszustand STOP ... A-4
A.1.3 Betriebszustand ANLAUF ... A-5
A.1.4 Betriebszustand RUN ... A-11
A.1.5 Betriebszustand HALT ... A-12
A.2 Speicherbereiche von S7-CPUs ... A-13
A.2.1 Aufteilung der Speicherbereiche ... A-13
A.2.2 Lade- und Arbeitsspeicher ... A-14
A.2.3 Systemspeicher ... A-16
A.2.3.1 Verwenden der Systemspeicherbereiche ... A-16
A.2.3.2 Prozeßabbild der Ein-/Ausgänge ... A-18
A.2.3.3 Lokaldaten-Stack ... A-21
A.2.3.4 Unterbrechungs-Stack ... A-22
A.2.3.5 Baustein-Stack ... A-22
A.2.3.6 Auswertung des Diagnosepuffers ... A-24
A.2.3.7 Remanente Speicherbereiche in S7-300-CPUs ... A-26
A.2.3.8 Remanente Speicherbereiche in S7-400-CPUs ... A-27
A.2.3.9 Konfigurierbare Speicherobjekte im Arbeitsspeicher ... A-28
A.3 Daten- und Parametertypen ... A-29
A.3.1 Einführung zu Daten- und Parametertypen ... A-29
A.3.2 Elementare Datentypen ... A-30
A.3.2.1 Format des Datentyps INT (16-Bit-Ganzzahlen) ... A-31
A.3.2.2 Format des Datentyps DINT (32-Bit-Ganzzahlen) ... A-32
A.3.2.3 Format des Datentyps REAL (Gleitpunktzahlen) ... A-32
A.3.2.4 Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen ... A-37
A.3.2.5 Format des Datentyps S5TIME (Zeitdauer) ... A-38
A.3.3 Zusammengesetzte Datentypen ... A-39
A.3.3.1 Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit) ... A-40
A.3.3.2 Verwenden von zusammengesetzten Datentypen ... A-41
A.3.3.3 Verwenden von Feldern für den Datenzugriff ... A-42
A.3.3.4 Verwenden von Strukturen für den Datenzugriff ... A-45
A.3.3.5 Verwenden von anwenderdefinierten Datentypen für den Datenzugriff ... A-47
A.3.4 Parametertypen ... A-49
A.3.4.1 Format der Parametertypen BLOCK, COUNTER, TIMER ... A-50
A.3.4.2 Format des Parametertyps POINTER ... A-51
A.3.4.3 Verwenden des Parametertyps POINTER ... A-52
A.3.4.4 Baustein zum Verändern des Pointers ... A-53
A.3.4.5 Format des Parametertyps ANY ... A-56
A.3.4.6 Verwenden des Parametertyps ANY ... A-58
A.3.4.7 Zuordnen von Datentypen zu Lokaldaten von Codebausteinen ... A-62
A.3.1.8 Zulässige Datentypen beim Übergeben von Parametern ... A-64
A.3.1.9 Übergabe an IN_OUT-Parameter eines FB ... A-69
A.4 Arbeiten mit älteren Projekten ... A-70
A.4.1 Umsetzen von Version 1-Projekten ... A-70
A.4.2 Umsetzen von Version 2-Projekten ... A-71
A.4.3 Hinweise zu STEP 7 V2.1-Projekten mit GD-Kommunikation ... A-72
A.4.4 Erweitern von DP-Slaves, die mit Vorgänger-Versionen von STEP 7 erstellt wurden ... A-72
A.4.5 DP-Slaves mit fehlenden oder fehlerhaften GSD-Dateien ... A-73
A.5 Beispielprogramme ... A-74
A.5.1 Beispielprojekte und Beispielprogramme ... A-74
A.5.2 Beispielprogramm für einen industriellen Mischprozeß ... A-75

A.5.2.1 Definieren von Codebausteinen A-78
A.5.2.2 Zuordnen von symbolischen Namen A-79
A.5.2.3 Erstellen des FB für den Motor A-81
A.5.2.4 Erstellen der FC für die Ventile A-85
A.5.1.5 Erstellen des OB 1 ..... A-87
A.5.3 Beispiel zur Hantierung mit Uhrzeitalarmen A-92
A.5.3.1 Struktur des Anwenderprogramms Uhrzeitalarme A-92
A.5.3.2 FC 12 ..... A-94
A.5.3.3 OB 10 ..... A-96
A.5.3.4 OB 1 und OB 80 ..... A-98
A.5.4 Beispiel zur Hantierung mit Verzögerungsalarmen A-99
A.5.4.1 Struktur des Anwenderprogramms Verzögerungsalarme A-99
A.5.4.2 OB 20 ..... A-101
A.5.4.3 OB 1 ..... A-102
A.5.5 Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen A-104
A.5.5.1 Beispiel zum Sperren und Freigeben von Alarm- und Asynchronereignissen (SFC 39 und 40) A-108
A.5.6 Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronereignissen (SFC 41 und 42) A-109
A.6 Zugriff auf Prozeß- und Peripheriedatenbereiche A-110
A.6.1 Zugriff auf den Prozeßdatenbereich A-110
A.6.2 Zugriff auf den Peripheriedatenbereich A-111
A.7 Einstellen des Betriebsverhaltens A-114
A.7.1 Einstellen des Betriebsverhaltens A-114
A.7.2 Ändern des Verhaltens und der Eigenschaften von Baugruppen A-115
A.7.3 Nutzen der Uhrzeitfunktionen A-117
A.7.4 Verwenden von Taktmerkern und Zeiten A-118

Index **Index-1**

1 Produkteinführung

1.1 Leitfaden durch STEP 7

Was ist STEP 7?

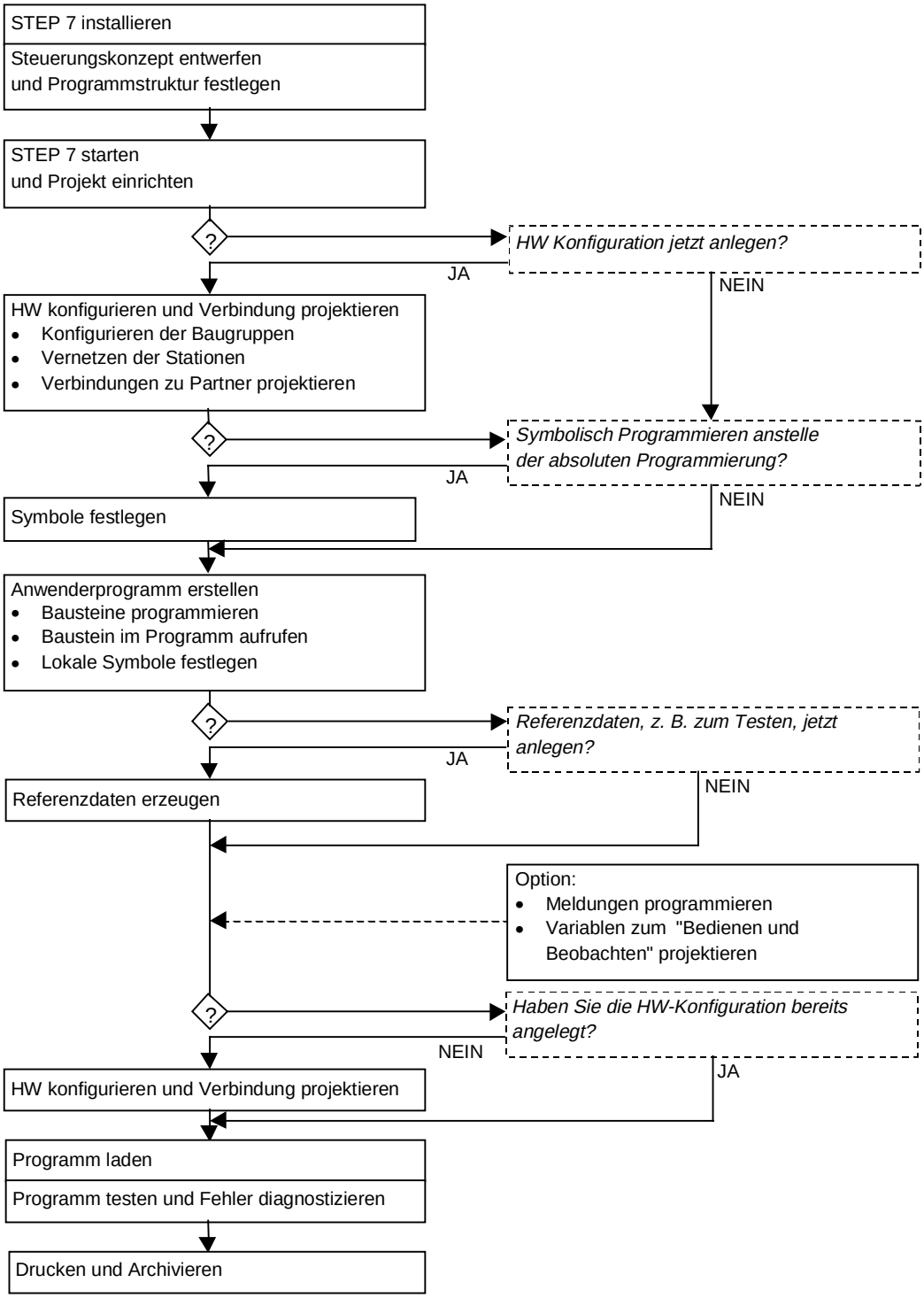
STEP 7 ist das Basispaket zur Konfigurierung und Programmierung von SIMATIC-Automatisierungssystemen. Es ist Teil der SIMATIC Industrie Software. Vom Basispaket STEP 7 gibt es folgende Ausführungen:

- STEP 7-Micro/DOS und STEP 7-Micro/Win für einfachere Stand-alone-Anwendungen auf SIMATIC S7-200.
- STEP 7 für Anwendungen auf SIMATIC S7-300/400, SIMATIC M7-300/400 und SIMATIC C7 mit zusätzlichem Funktionsumfang:
  - optional erweiterbar durch die in SIMATIC Industrie Software enthaltenen Softwareprodukte (siehe auch Erweiterungsmöglichkeiten des STEP 7-Basispakets)
  - Möglichkeit, Funktionsbaugruppen und Kommunikationsbaugruppen zu parametrieren
  - Forcen und Multicomputing-Betrieb
  - Globaldaten-Kommunikation
  - ereignisgesteuerte Datenübertragung mit Kommunikations-/Funktionsbausteinen
  - Projektieren von Verbindungen

STEP 7 ist Gegenstand dieser Dokumentation, STEP 7-Micro ist in der Dokumentation zu "STEP 7-Micro/DOS" beschrieben.

Grundlegende Arbeiten

Bei der Erstellung einer Automatisierungslösung mit STEP 7 fallen grundlegende Aufgaben an. Nachfolgendes Bild zeigt die Aufgaben, die in den meisten Projekten durchgeführt werden, und ordnet sie zu einer prinzipiellen Vorgehensweise in Form eines Leitfadens. Er verweist dabei auf die jeweiligen Kapitel und gibt Ihnen damit die Möglichkeit, sich aufgabenspezifisch durch das Handbuch zu bewegen.


STEP 7 installieren
Steuerungskonzept entwerfen
und Programmstruktur festlegen
STEP 7 starten
und Projekt einrichten
?
HW Konfiguration jetzt anlegen?
JA
NEIN
HW konfigurieren und Verbindung projektieren
• Konfigurieren der Baugruppen
• Vernetzen der Stationen
• Verbindungen zu Partner projektieren
?
Symbolisch Programmieren anstelle der absoluten Programmierung?
JA
NEIN
Symbole festlegen
Anwenderprogramm erstellen
• Bausteine programmieren
• Baustein im Programm aufrufen
• Lokale Symbole festlegen
?
Referenzdaten, z. B. zum Testen, jetzt anlegen?
JA
NEIN
Referenzdaten erzeugen
Option:
• Meldungen programmieren
• Variablen zum "Bedienen und Beobachten" projektieren
?
Haben Sie die HW-Konfiguration bereits angelegt?
NEIN
JA
HW konfigurieren und Verbindung projektieren
Programm laden
Programm testen und Fehler diagnostizieren
Drucken und Archivieren

Alternativen in der Vorgehensweise

Wie im vorherigen Bild dargestellt, haben Sie für die Vorgehensweise zwei Alternativen:

- Sie können zuerst die Hardware konfigurieren und danach die Bausteine programmieren.
- Sie können aber auch zuerst die Bausteine programmieren, ohne vorher die Hardware konfigurieren zu müssen. Das empfiehlt sich vor allem bei Service- und Wartungsarbeiten, um z. B. programmierte Bausteine in ein bereits bestehendes Projekt zu integrieren.

Kurzbeschreibung der einzelnen Arbeitsschritte:

- Installieren und Autorisieren
  Bei der Erstanwendung installieren Sie STEP 7 und übertragen die Autorisierung von Diskette auf die Festplatte (siehe auch Installieren von STEP 7 und Autorisierung).
- Steuerung entwerfen
  Bevor Sie mit STEP 7 arbeiten, planen Sie Ihre Automatisierungslösung von der Untergliederung des Prozesses in einzelne Aufgaben bis zur Erstellung eines Konfigurationsplanes (siehe auch Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung).
- Programmstruktur entwerfen
  Die beim Steuerungsentwurf beschriebenen Aufgaben setzen Sie mit den von STEP 7 zur Verfügung gestellten Bausteinen in eine Programmstruktur um (siehe auch Bausteine im Anwenderprogramm).
- STEP 7 starten
  Sie starten STEP 7 von der Windows-Oberfläche aus (siehe auch Starten von STEP 7).
- Projektstruktur anlegen
  Ein Projekt ist wie ein Ordner, in dem alle Daten hierarchisch gegliedert abgelegt werden und jederzeit zur Verfügung stehen. Nachdem Sie ein Projekt angelegt haben, werden alle weiteren Arbeiten in diesem Projekt ausgeführt (siehe auch Projektstruktur).
- Station einrichten
  Mit dem Einrichten der Station legen Sie die Steuerung fest: z. B. SIMATIC 300, SIMATIC 400, SIMATIC S5 (siehe auch Einfügen von Stationen).

- Hardware konfigurieren
  Beim Konfigurieren legen Sie in einer Konfigurationstabelle fest, welche Baugruppen Sie für Ihre Automatisierungslösung einsetzen und über welche Adressen die Baugruppen aus dem Anwenderprogramm heraus angesprochen werden sollen. Darüber hinaus lassen sich noch Eigenschaften der Baugruppen über Parameter einstellen (siehe auch Grundsätzliche Bedienung beim Hardware konfigurieren).
- Netze und Kommunikationsverbindungen projektieren
  Grundlage für die Kommunikation ist ein zuvor konfiguriertes Netz. Dazu erzeugen Sie die für Ihre Automatisierungsnetze benötigten Subnetze, legen die Subnetz-Eigenschaften und für die vernetzten Stationen die Netzanschluß-Eigenschaften und ggf. die benötigten Kommunikationsverbindungen fest (siehe auch Vorgehensweise zum Projektieren eines Subnetzes).
- Symbole festlegen
  Sie können in einer Symboltabelle anstelle von Adressen lokale oder globale Symbole mit einer aussagekräftigen Bezeichnung definieren, um sie dann in Ihrem Programm zu verwenden (siehe auch Anlegen einer Symboltabelle)
- Programm erstellen
  Mit einer der zur Verfügung stehenden Programmiersprachen erstellen Sie ein baugruppenzugeordnetes oder baugruppenunabhängiges Programm und hinterlegen dieses als Bausteine, Quellen oder Pläne (siehe auch Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen und Grundlagen zum Programmieren in AWL-Quellen).
- Nur S7: Referenzdaten erzeugen und auswerten
  Sie können sich dieser Referenzdaten bedienen, um sich das Testen und Ändern Ihres Anwenderprogramms zu erleichtern (siehe auch Übersicht der möglichen Referenzdaten).
- Meldungen projektieren
  Sie legen z. B. bausteinbezogene Meldungen mit ihren Texten und Attributen an. Mit dem Transferprogramm transferieren Sie die erstellten Meldungsprojektierungsdaten in den Datenbestand des Bedien- und Beobachtungssystems (z.B. SIMATIC WinCC, SIMATIC ProTool), siehe auch Projektieren von Meldungen.
- B+B-Variable projektieren
  Sie legen einmalig in STEP 7 B+B-Variablen an und statten diese mit den gewünschten Attributen aus. Mit dem Transferprogramm transferieren Sie die erstellten B+B-Variablen in den Datenbestand des Bedien- und Beobachtungssystems WinCC (siehe auch Projektieren von bedien- und beobachtbaren Variablen).

- Programme auf Zielsystem laden
  Nur S7: Nach Abschluß von Konfiguration, Parametrierung und Programmerstellung können Sie Ihr komplettes Anwenderprogramm oder einzelne Bausteine davon auf das Zielsystem (programmierbare Baugruppe Ihrer Hardwarelösung) übertragen (siehe auch Voraussetzungen für das Laden). Die CPU enthält bereits das Betriebssystem.
  Nur M7: Unter verschiedenen Betriebssystemen wählen Sie das für Ihre Automatisierungslösung passende aus und übertragen dieses einzeln oder zusammen mit dem Anwenderprogramm auf den gewünschten Datenträger des M7-Zielsystems.
- Programme testen
  Nur S7: Zum Testen haben Sie die Möglichkeit, sich Werte von Variablen aus Ihrem Anwenderprogramm oder einer CPU anzeigen zu lassen, diesen Werte zuzuweisen und eine Variablentabelle für die Variablen zu erstellen, die Sie anzeigen oder steuern wollen (siehe auch Einführung zum Testen mit der Variablentabelle).
  Nur M7: Testen des Anwenderprogramms mit einem Hochsprachen-Debugger.
- Betrieb überwachen, Hardware diagnostizieren
  Sie ermitteln die Ursache einer Baugruppenstörung, indem Sie sich Online-Informationen zu einer Baugruppe anzeigen lassen. Die Ursache für die Störung im Ablauf eines Anwenderprogramms ermitteln Sie mit Hilfe des Diagnosepuffers und der Stack-Inhalte. Darüber hinaus können Sie prüfen, ob ein Anwenderprogramm auf einer bestimmten CPU ablauffähig ist (siehe auch Diagnose der Hardware und Anzeigen des Baugruppenzustands).
- Anlage dokumentieren
  Nach dem Erstellen eines Projektes/einer Anlage ist es sinnvoll, die Projektdaten übersichtlich zu dokumentiert, um die weitere Bearbeitung des Projektes als auch Servicearbeiten zu erleichtern (siehe auch Drucken einer Projektdokumentation). DOCPRO, das optionale Werkzeug zum Erstellen und Verwalten von Anlagendokumentation, ermöglicht die Strukturierung der Projektdaten, die Aufbereitung in Form von Schaltbüchern und den Ausdruck in einem einheitlichen Druckbild.

Erweiterung des Leitfadens um spezielle Themen

Bei der Erstellung einer Automatisierungslösung gibt es eine Reihe von speziellen Themen, die für Sie von Interesse sein können:

- Multicomputing - Synchroner Betrieb mehrerer CPUs (siehe auch Multicomputing - Synchroner Betrieb mehrerer CPUs)
- Arbeiten von mehreren Personen in einem Projekt (siehe auch Bearbeiten von Projekten durch mehrere Personen)
- Arbeiten mit M7-Systemen (siehe auch Vorgehensweise für M7-Systeme)

1.2 Das STEP 7 - Basispaket

Verwendete Standards

Die in STEP 7 integrierten SIMATIC-Programmiersprachen erfüllen die Norm DIN EN 6.1131-3. Das Basispaket läuft unter dem Betriebssystem Windows 95/98/NT/2000 und ist an dessen grafische und objektorientierte Arbeitsweise angepaßt.

Funktionen des Basispakets

Das Basispaket unterstützt Sie in allen Phasen des Erstellungsprozesses von Automatisierungslösungen, wie z.B.:

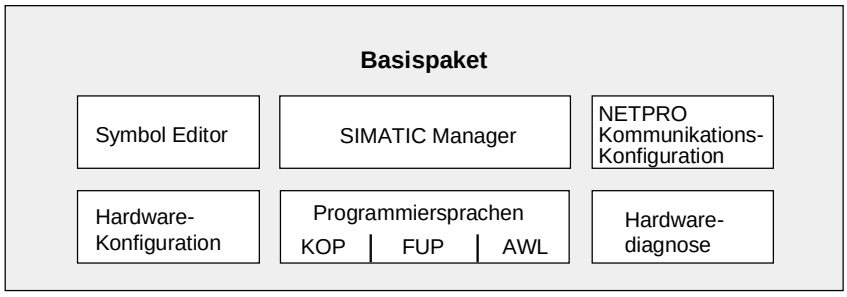
- Einrichten und Verwalten von Projekten,
- Konfigurieren und Parametrieren der Hardware und der Kommunikation,
- Verwaltung von Symbolen,
- Programmerstellung z. B. für S7-Zielsysteme,
- Laden von Programmen auf Zielsysteme,
- Test der Automatisierungsanlage,
- Diagnose bei Anlagenstörungen.

Die Benutzungsoberfläche der STEP 7-Software ist nach modernen ergonomischen Erkenntnissen gestaltet und ermöglicht einen leichten Einstieg.

Die Dokumentation zum Softwareprodukt STEP 7 bietet Ihnen die gesamte Information online in der Online-Hilfe und in elektronischen Handbüchern im PDF-Format.

Unterstützende Tools

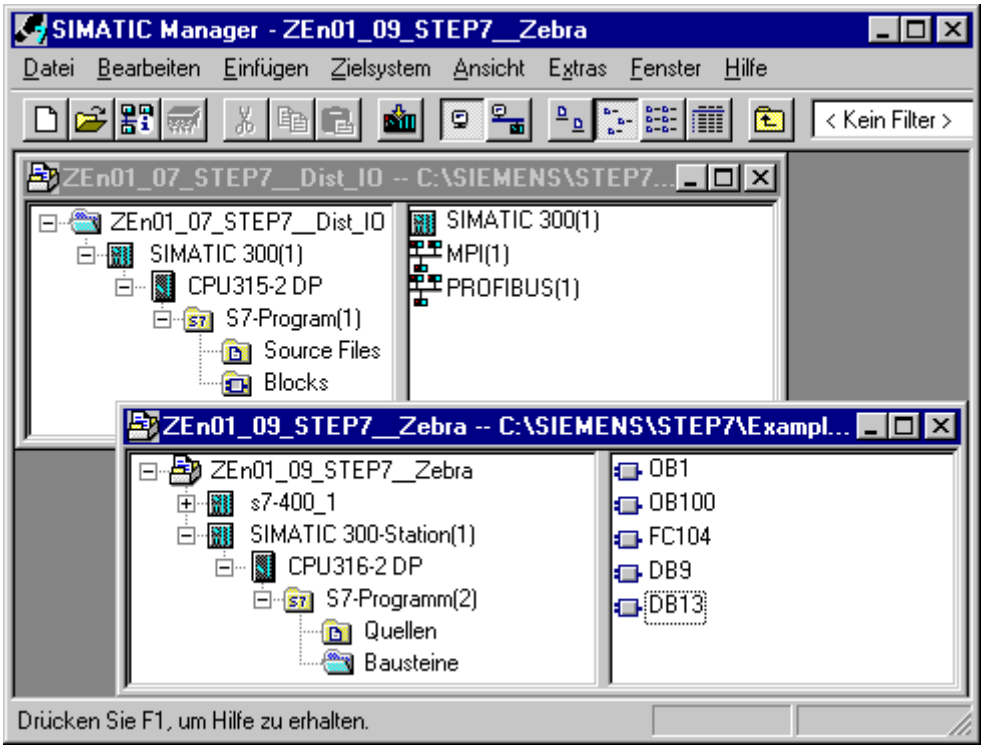
Das STEP 7-Basispaket stellt Ihnen eine Reihe von Tools zur Verfügung:



Die Tools müssen Sie nicht separat aufrufen, sie werden automatisch gestartet, wenn Sie eine zugehörige Funktion anwählen oder ein Objekt öffnen.

SIMATIC Manager

Der SIMATIC Manager verwaltet alle Daten, die zu einem Automatisierungsprojekt gehören – unabhängig davon, auf welchem Zielsystem (S7/M7/C7) sie realisiert sind. Die Tools, die zur Bearbeitung der angewählten Daten erforderlich sind, werden automatisch vom SIMATIC Manager gestartet.



Symbol Editor

Mit dem Symbol Editor verwalten Sie alle globalen Variablen. Es stehen folgende Funktionen zur Verfügung:

- Festlegen von symbolischen Bezeichnungen und Kommentaren zu den Prozeßsignalen (Ein-/Ausgänge), Merkern und Bausteinen,
- Sortierfunktionen,
- Import/Export mit anderen Windows-Programmen.

Die dabei entstehende Symboltabelle steht allen Werkzeugen zur Verfügung. Die Änderung eines Symbolparameters wird deshalb automatisch von allen Tools erkannt.

Hardware-Diagnose

Die Hardware-Diagnose bietet Ihnen einen Überblick über den Zustand des Automatisierungssystems. In einer Übersichtsdarstellung kann für jede Baugruppe anhand eines Symbols angezeigt werden, ob sie gestört ist oder nicht. Durch Doppelklick auf die gestörte Baugruppe werden detaillierte Informationen zur Störung angezeigt. Der Umfang dieser Informationen ist abhängig von der einzelnen Baugruppe:

- Anzeige allgemeiner Informationen zur Baugruppe (z.B. Bestellnummer, Version, Bezeichnung) und des Zustands der Baugruppe (z. B. gestört),
- Anzeige der Baugruppenfehler (z. B. Kanalfehler) von zentraler Peripherie und DP-Slaves,
- Anzeige der Meldungen aus dem Diagnosepuffer.

Für CPUs wird zusätzlich Information angezeigt:

- Ursachen für Störung im Ablauf eines Anwenderprogramms,
- Anzeige der Zyklusdauer (längster, kürzester und letzter Zyklus),
- Möglichkeiten und Auslastung der MPI-Kommunikation,
- Anzeige der Leistungsdaten (Anzahl möglicher Ein-/Ausgänge, Merker, Zähler, Zeiten und Bausteine).

Programmiersprachen

Die Programmiersprachen KOP, AWL und FUP für S7-300/400 sind integraler Bestandteil des Basispakets.

- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Syntax der Anweisungen ähnelt einem Stromlaufplan. KOP ermöglicht Ihnen eine einfache Verfolgung des Signalflusses zwischen Stromschienen über Kontakte, komplexe Elemente und Spulen.
- AWL (Anweisungsliste) ist eine textuelle, maschinennahe Programmiersprache. Wird ein Programm in AWL programmiert, so entsprechen die einzelnen Anweisungen weitgehend den Arbeitsschritten, mit denen die CPU das Programm bearbeitet. Zur Erleichterung der Programmierung wurde AWL um einige Hochsprachenkonstrukte (wie z.B. strukturierte Datenzugriffe und Bausteinparameter) erweitert.
- FUP (Funktionsplan) ist eine grafische Programmiersprache und benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten logischen Boxen. Außerdem können komplexe Funktionen (z.B. mathematische Funktionen) direkt in Verbindung mit den logischen Boxen dargestellt werden.

Weitere Programmiersprachen sind als Optionssoftware erhältlich.

Hardware-Konfiguration

Dieses Tool setzen Sie zur Konfiguration und Parametrierung der Hardware eines Automatisierungsprojektes ein. Es stehen folgende Funktionen zur Verfügung:

- Um das Automatisierungssystem zu konfigurieren, wählen Sie Baugruppenträger (Racks) aus einem elektronischen Katalog aus und ordnen die ausgewählten Baugruppen den gewünschten Steckplätzen in den Racks zu.
- Die Konfigurierung der Dezentralen Peripherie erfolgt identisch zur Konfiguration der zentralen Peripherie. Dabei wird auch kanalgranulare Peripherie unterstützt.
- Bei der Parametrierung der CPU können Sie Eigenschaften wie Anlaufverhalten und Zykluszeitüberwachung menügeführt einstellen. Multicomputing wird unterstützt. Die eingegebenen Daten werden in Systemdatenbausteinen abgelegt.
- Bei der Parametrierung der Baugruppen lassen sich alle einstellbaren Parameter über Dialogfelder festlegen. Einstellungen über DIP-Schalter entfallen. Die Parametrierung der Baugruppen erfolgt automatisch im Hochlauf der CPU. So ist z. B. ein Baugruppentausch ohne erneute Parametrierung möglich.
- Die Parametrierung von Funktionsmodulen (FMs) und Kommunikationsprozessoren (CPs) erfolgt ebenso innerhalb der Hardwarekonfiguration identisch zur Parametrierung der restlichen Baugruppen. Zu jedem FM und CP werden hierzu baugruppenspezifische Dialogfelder und Regeln zur Verfügung gestellt (im Lieferumfang FM-/CP-Funktionspaket). Das System verhindert fehlerhafte Eingaben, indem in den Dialogfeldern nur zulässige Eingabemöglichkeiten angeboten werden.

NetPro

Durch NetPro ist eine zeitgesteuerte zyklische Datenübertragung über MPI möglich mit:

- Auswahl der Kommunikationsteilnehmer,
- Eintragen von Datenquelle und Datenziel in eine Tabelle; das Generieren aller zu ladenden Bausteine (SDB) und ihr vollständiges Übertragen auf alle CPUs erfolgt automatisch.

Darüber hinaus ist eine ereignisgesteuerte Datenübertragung möglich mit:

- Festlegen der Kommunikationsverbindungen,
- Auswahl der Kommunikations-/Funktionsbausteine aus der integrierten Bausteinbibliothek,
- Parametrierung der ausgewählten Kommunikations-/Funktionsbausteine in der gewohnten Programmiersprache.

1.3 Was ist neu in STEP 7 Version 5.1

SIMATIC Manager

- Für die Übersetzung von Projekten in andere Landessprachen können Sie Texte des Projekts (z. B. Bausteintitel und Kommentare) über die Menübefehle **Extras > Texte mehrsprachig verwalten > Exportieren** außerhalb von STEP 7 mit einem ASCII-Editor oder einem Tabellenbearbeitungswerkzeug bearbeiten und anschließend mit dem Menübefehl **Extras > Texte mehrsprachig verwalten > Importieren** wieder in STEP 7 importieren. Das Format der Export-Datei ist festgelegt auf "*.csv" (comma separated value).
- Die kompletten Projektdaten können in dafür geeignete Memory Cards einer CPU geladen werden (neue Menübefehle **Zielsystem > Projekt auf Memory Card speichern** und **Zielsystem > Projekt aus Memory Card holen**)
- Mit dem Menübefehl **Extras > Referenzdaten > Löschen** können Sie die vorhandenen Referenzdaten löschen.
- Die Versionsinformationen zu installierten Produkten mit ihren Komponenten und DLLs können über den Menübefehl **Hilfe > Info** ausgelesen werden.
- Mit dem Menübefehl **Bearbeiten > Bausteinkonsistenz prüfen** können Sie über alle S7-Bausteine im Baustein-Ordner nach Programmänderung eine Konsistenzprüfung anstoßen. So können Sie Auswirkungen von Schnittstellenänderungen auf andere Bausteine besser kontrollieren und Fehler schneller beheben.
- Systemattribute, die Sie bereits für Bausteine Ihres Anwenderprogramms definiert haben, können beim Import neuer Bausteinversionen übernommen werden (z. B. beim Hochrüsten einer Systembibliothek). Über einen Dialog können Sie die Attribute für jeden Baustein abgleichen.

KOP/AWL/FUP-Bausteine programmieren

- Mit dem neuen Menübefehl **Datei > Zugriffe überprüfen und aktualisieren** wird die Bausteinkonsistenzprüfung gestartet.
- Es besteht die Möglichkeit, die aufgerufenen Bausteine mit **"Beobachten"** oder mit **"Beobachten mit Aufrufpfad"** zu beobachten, wenn der Betriebsmodus auf "Testbetrieb" steht. Dazu müssen Sie den aufrufenden Baustein öffnen und den Cursor auf den gewünschten Aufruf (CALL-Zeile in AWL und Callbox in KOP/FUP) positionieren. Mit der rechten Maustaste wählen Sie dann zwischen den Befehlen **Aufgerufenen Baustein > Beobachten** und **Aufgerufenen Baustein > Beobachten mit Aufrufpfad**.
- Beim Löschen eines Bausteins wird das Symbol des Bausteins mitgelöscht. Dies hat zur Folge, daß symbolisch führende Quellen nicht mehr übersetzt werden können, wenn die zugehörigen Bausteine aus dem Programm gelöscht wurden.
  Beim Kopieren und Verschieben eines Bausteins bleibt das Symbol erhalten.

Variable beobachten/steuern

- Die Tabelle für das Beobachten und Steuern von Variablen wurde überarbeitet:
  - Spalten können selektiert werden
  - Mehrfachselekton möglich
  - Alle Spalten sind ein- und ausblendbar
  - Tooltip wird angeboten z. B. für rote Zeile
  - Das Anzeigeformat ist editierbar.
- Dialogfeld "Einstellungen" mit zwei neuen Registern ("Allgemein" und "Online"); über das Register "Online" sind folgende neue Optionen wählbar:
  - Vorauswahl für die Online-Verbindung: entweder zur direkt angeschlossenen CPU oder zur projektierten CPU
  - Ausschalten von Warnhinweisen
  - Option "Variablen zusammenfassen" zur Vergrößerung der Anzahl max. beobachtbarer Variablen
- Wechseln der Verbindung ist möglich, ohne vorher die bestehende Verbindung abgebaut zu haben
- Trigger für Beobachten ist während des Beobachtens von Variablen einstellbar
- Ausgewählte Variablen können gesteuert werden, indem Sie die betreffenden Zeilen markieren und die Funktion "Steuern" ausführen. Nur die sichtbaren markierten Variablen werden dann gesteuert.
- Zahlreiche neue Menübefehle: u. a.
  - Druckvorschau (Menü "Tabelle")
  - Anordnung wiederherstellen (Menü "Fenster")
  - Verbindung herstellen zu 1, 2 ,3 ,4 (Menü "Zielsystem"; zum schnellen Wechsel bereits benutzter Verbindungen)

Hardware konfigurieren und diagnostizieren

- Beobachten/Steuern von Ein-/Ausgängen beim Hardware konfigurieren möglich (neuer Menübefehl **Zielsystem > Beobachten/Steuern**)
- Neue Baugruppen, z. B. IM 151/CPU als intelligenter DP-Slave aus der Familie ET 200S
- Erweiterte Konfigurationsmöglichkeiten bei intelligenten DP-Slaves:
  Teilprozeßabbild-Zuordnung für S7-400-CPU mit direktem Datenaustausch
  und
  Prozeßalarm-OB-Zuordnung für den PROFIBUS-Partner (für I-Slaves, die anwenderprogramm-gesteuert beim DP-Master einen Prozeßalarm auslösen können)
- Ergonomische Verbesserungen für die Online-Funktion "Baugruppenzustand":
  Für das Register "Diagnosepuffer" können die Ereignisse gefiltert angezeigt werden (einzelne Ereignisklassen ausblenden)
  Im Register "Leistungsdaten" sind jetzt Informationen zu Organisationsbausteinen, Systemfunktionen (SFCs und SFBs) sowie zu

Operandenbereichen zusammengefaßt. Die Informationen zum Speicher sind komplett im Register "Speicher" zu finden.
Die grafische Darstellung zur Zykluszeit mit den entsprechenden Überwachungszeiten ist durch eine waagerechte Anordnung der Zeitachse verbessert worden. In dieser Darstellung sind Über- und Unterschreitungen der parametrierten Überwachungszeiten leichter erkennbar.

Netze und Verbindungen projektieren

- Neue Spalten in der Verbindungstabelle: Lokale und Partner Schnittstelle sowie Lokale und Partner Adresse. Die Spalten lassen sich einzeln ein-bzw. ausblenden. Auf diese Weise läßt sich der Verbindungsweg vollständig aus der Verbindungstabelle ablesen und z. B. nach Schnittstellen oder Subnetzen sortieren.
- Die Einstellungen in NetPro werden beim Beenden des Projekts gespeichert und stehen beim erneuten Öffnen (auch auf einem anderen PG) zur Verfügung.
- Die Subnetze können besser auseinander gehalten werden, da sie auf dem Bildschirm in unterschiedlicher Farbe dargestellt werden.
  Im Dialog für die Druck-Einstellungen können bei grafischem Ausdruck die Farben ausgeschaltet werden.
  Außerdem kann eine Zoom-Einstellung vorgenommen werden, so daß der Ausdruck der Netzdarstellung angepaßt werden kann, um die zur Verfügung stehende Seitenanzahl optimal auszunutzen.
- Neben den Busparametern für PROFIBUS können Busparameter für andere Subnetze (MPI) ausgedruckt werden.
- Unterstützung der Verbindungsprojektierung (S7-Verbindungen) und Verbindungsstatus für die neuen WinAC-SlotCPUs (CPU 41x-2 DP PCI)

Referenzdaten

- Mit dem Menübefehl **Bearbeiten > Symbole löschen** können Sie in der Ansicht "Nicht verwendete Symbole" die nicht verwendeten Symbole löschen.
- Mit dem Menübefehl **Bearbeiten > Symbole bearbeiten** können Sie in der Ansicht "Operanden ohne Symbol" den selektierten Operanden Symbole zuordnen.
- Die Anordnung der Fenster wird unabhängig von der angezeigten Ansicht (Querverweise, Programmstruktur,...) beim Beenden der Applikation gespeichert und wiederhergestellt, wenn der Menübefehl **Fenster > Anordnung beim Beenden speichern** gewählt ist.

Projektieren von Meldungen

- Sie können anwenderdefinierte Diagnosemeldungen auch für M7-Programme anlegen.
- Das Dialogfeld "PCS7-Meldungsprojektierung" für das Bearbeiten eines ereignisgesteuerten Kommunikationsbausteins enthält zwei Register in denen bis zu 10 Meldetexte eingegeben werden können.

CPU Melden

- Sie können die an der Applikation CPU-Melden ankommenden Meldungen mit verschiedenen Optionen bearbeiten:
  Mit dem Menübefehl **Ansicht > Automatisches Verschieben** werden immer die neu ankommenden Meldungen in das Fenster gescrollt und selektiert.
  Mit dem Menübefehl **Ansicht > In den Vordergrund holen** wird das Fenster in den Vordergrund geblendet und die Meldung angezeigt.
  Mit dem Menübefehl **Ansicht > Im Hintergrund lassen** werden Meldungen im Fenster angezeigt, aber das Fenster bleibt im Hintergrund.
  Mit dem Menübefehl **Ansicht > Meldung ignorieren** werden Meldungen weder im Fenster angezeigt noch im Archiv gespeichert.
- Sie können die markierte Baugruppe mit dem Menübefehl **Zielsystem > Baugruppe entfernen** aus der Liste entfernen
- Sie können mit dem Dialogfeld "Einstellungen-CPU-Melden" die Größe des Archivs einstellen, die Liste der angemeldeten Baugruppen speichern und den Verbindungszustand beim Start wiederherstellen lassen. Sie können sich außerdem die Infotexte bei ALARM S/SQ anzeigen lassen.

Systemfehler melden

- STEP 7 bietet mit der Funktion "Systemfehler melden" eine komfortable Möglichkeit, die von der Komponente zur Verfügung gestellten Diagnoseinformationen in Form von Meldungen anzuzeigen. Die hierfür notwendigen Bausteine und Meldetexte werden von STEP 7 automatisch erzeugt. Sie müssen die erzeugten Bausteine lediglich in die CPU laden und die Texte in angeschlossene HMI-Geräte transferieren. Eine genaue Übersicht über die unterstützten Diagnose-Informationen bei verschiedenen DP-Slaves finden Sie unter "Unterstützte Komponenten und Funktionsumfang".

1.4 Erweiterungsmöglichkeiten des STEP 7 - Basispakets

1.4.1 Erweiterungsmöglichkeiten des STEP 7 - Basispakets

Das Basispaket kann durch optionale Software, die in den drei folgenden Softwareklassen zusammengefaßt sind, erweitert werden.

- Engineering Tools;
  diese umfassen höhere Programmiersprachen und technologieorientierte Software.
- Runtime Software;
  enthält fertige Ablaufsoftware für den Produktionsprozeß.
- Human Machine Interfaces (HMI);
  steht für Software speziell zum Bedienen und Beobachten.

Nachfolgende Tabelle zeigt die einsetzbare Optionssoftware in Abhängigkeit von Automatisierungssystem:

		STEP 7	
	S7-300 S7-400	**M7-300 M7-400**	**C7-620**
Engineering Tools			
• Borland C/C++		o	
• CFC	+[1)]	+	+[2)]
• DOCPRO	+	+[3)]	+
• HARDPRO	+		
• M7-ProC/C++		o	
• S7-GRAPH	+[1)]		+[2)]
• S7-HiGraph	+		+
• S7-PDIAG	+		
• S7-PLCSIM	+		+
• S7-SCL	+		+
• TeleService	+	+	+
Runtime Software			
• Fuzzy Control	+		+
• M7-DDE-Server		+	
• M7-SYS RT		o	
• Modular PID Control	+		+
• PC-DDE-Server	+		
• PRODAVE MPI	+		
• Standard PID Control	+		+
Human Machine Interface			
• ProAgent			
• SIMATIC ProTool			
• SIMATIC ProTool/Lite			o

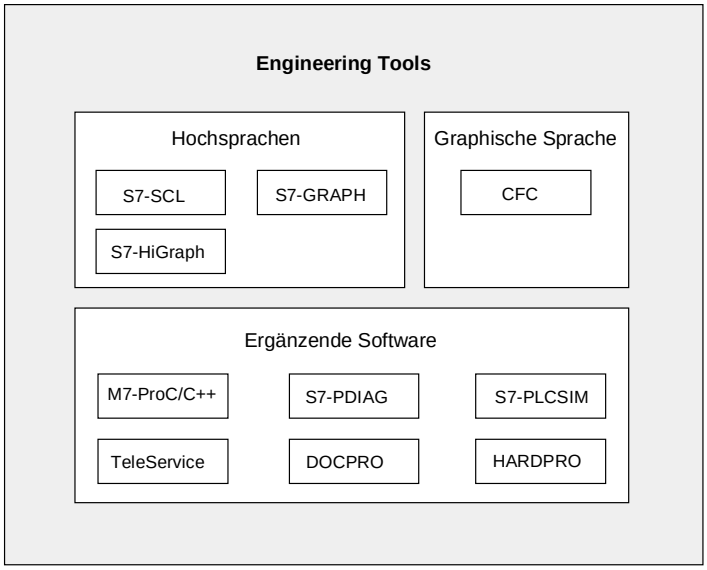
		STEP 7	
	S7-300 **S7-400**	**M7-300** **M7-400**	**C7-620**
• SIMATIC WinCC			

o = unbedingt erforderlich
\+ = optional
[1)] = empfohlen ab S7-400
[2)] = nicht empfohlen für C7-620
[3)] = nicht für C-Programme

1.4.2 Engineering Tools

Engineering Tools sind aufgabenorientierte Tools, die als Erweiterung des Basispakets eingesetzt werden können. Engineering Tools umfassen:

- Hochsprachen für den Programmierer
- Grafische Sprache für den Technologen
- Ergänzende Software für Diagnose, Simulation, Fernwartung, Anlagendokumentation, etc.



Hochsprachen

Für die Programmierung der Automatisierungssysteme SIMATIC S7-300/400 stehen folgende Sprachen als Optionssoftware zur Verfügung:

- S7-GRAPH ist eine Programmiersprache zur komfortablen Beschreibung von Ablaufsteuerungen (Schrittkettenprogrammierung). Der Prozeßablauf wird dabei in Schritte zerlegt. Die Schritte enthalten insbesondere Aktionen zum Steuern der Ausgänge. Der Übergang von einem Schritt zum nächsten wird durch Weiterschaltbedingungen gesteuert.
- S7-HiGraph ist eine Programmiersprache zur komfortablen Beschreibung asynchroner, nicht sequentieller Vorgänge in Form von Zustandsgraphen. Dazu wird die Anlage in Funktionseinheiten gegliedert, die verschiedene Zustände einnehmen können. Die Funktionseinheiten können sich durch Austausch von Nachrichten synchronisieren.
- S7-SCL ist eine textuelle Hochsprache nach DIN EN 61131-3. Sie enthält Sprachkonstrukte, wie man sie in ähnlicher Form auch in den Programmiersprachen Pascal und C findet. S7-SCL eignet sich daher besonders für Benutzer, die schon gewöhnt sind, mit einer höheren Programmiersprache zu arbeiten. S7-SCL ist beispielsweise einsetzbar zur Programmierung häufig wiederkehrender oder sehr komplexer Funktionen.

Grafische Sprache

CFC für S7 und M7 ist eine Programmiersprache zur grafischen Verschaltung vorhandener Funktionen. Die Funktionen decken einen weiten Bereich von einfachen Logikverknüpfungen bis hin zu komplexen Regelungen und Steuerungen ab. Eine große Anzahl solcher Funktionen steht in Form von Bausteinen in einer Bibliothek zur Verfügung. Sie programmieren, indem Sie Bausteine auf einen Plan kopieren und Bausteinanschlüsse durch Linien verbinden.

Ergänzende Software

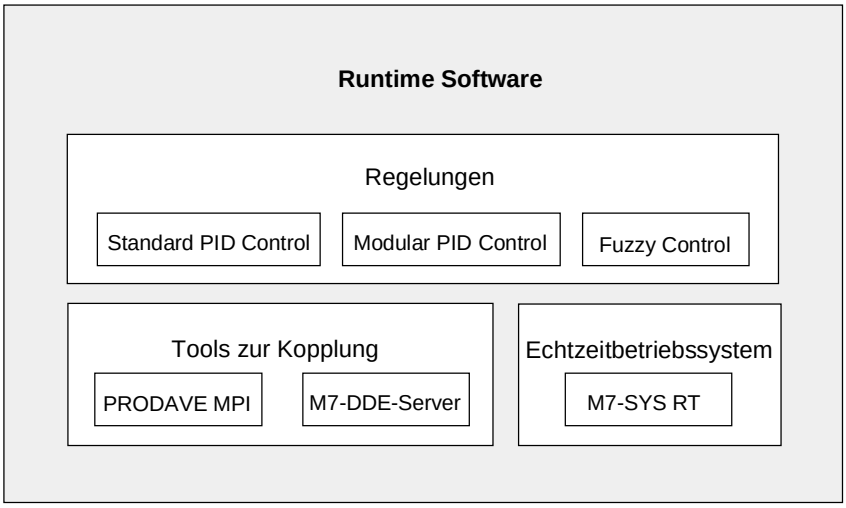
- Borland C++ (nur M7) enthält die Borland Entwicklungsumgebung.
- Mit DOCPRO können Sie alle Projektierungsdaten, die mit STEP 7 erstellt wurden, in Schaltbüchern organisieren. Schaltbücher ermöglichen das bequeme Verwalten der Projektierungsdaten und eine normgerechte Druckaufbereitung.
- HARDPRO ist das Hardwareprojektierungssystem für S7-300 zur Anwenderunterstützung bei umfangreichen Projektierungen für komplexe Automatisierungsaufgaben.
- M7-ProC/C++ (nur M7) ermöglicht die Einbindung der Borland-Entwicklungsumgebung für die Programmiersprachen C und C++ in die STEP 7-Entwicklungsumgebung.
- Mit S7-PLCSIM (nur S7) können Sie zu Testzwecken an das Erstellsystem (PC/PG) angeschlossene S7-Automatisierungssysteme simulieren.

- S7-PDIAG (nur S7) ermöglicht die einheitliche Projektierung der Prozeßdiagnose für SIMATIC S7-300/400. Mit Hilfe der Prozeßdiagnose lassen sich fehlerhafte Zustände außerhalb des Automatisierungssystems erkennen (z.B. Endschalter nicht erreicht).
- TeleService bietet die Möglichkeit S7- und M7-Automatisierungssysteme mit dem PG/PC über das Telefonnetz zu programmieren und zu warten.

1.4.3 Runtime Software

Runtime Software umfaßt fertig programmierte Lösungen, die durch das Anwenderprogramm aufgerufen werden. Runtime Software wird direkt in die Automatisierungslösung eingebaut. Sie umfaßt:

- Regelungen für SIMATIC S7, z. B. Standard-, Modular- und Fuzzyregelungen
- Tools zur Kopplung der Automatisierungssysteme mit Windows-Applikationen
- ein Echtzeitbetriebssystem für SIMATIC M7



Regelungen für SIMATIC S7

- Standard PID Control ermöglicht die Integration von kontinuierlichen Reglern, Impulsreglern und Schrittreglern in das Anwenderprogramm. Das Parametriertool mit integrierter Reglereinstellung erlaubt es, den Regler in kürzester Zeit zu parametrieren und optimal einzustellen.
- Modular PID Control kommt dann zum Einsatz, wenn ein einfacher PID-Regler zur Lösung der Automatisierungsaufgabe nicht ausreicht. Durch Verschaltung der mitgelieferten Standard-Funktionsbausteine läßt sich nahezu jede regelungstechnische Struktur realisieren.
- Mit Fuzzy Control können Fuzzy-Systeme erstellt werden. Diese Systeme kommen dann zum Einsatz, wenn Prozesse mathematisch nur schwer oder gar nicht zu beschreiben sind, Abläufe und Prozesse unvorhersehbar verlaufen, Nichtlinearitäten auftreten, andererseits aber Erfahrungswissen im Umgang mit dem Prozeß vorliegt.

Tools zur Kopplung

- PRODAVE MPI ist eine Toolbox für den Prozeßdatenverkehr zwischen SIMATIC S7, SIMATIC M7 und SIMATIC C7. Sie wickelt den Datenverkehr selbständig über die MPI-Schnittstelle ab.
- Mit dem M7-DDE-Server (**D**ynamic **D**ata **E**xchange) lassen sich Windows-Applikationen ohne zusätzlichen Programmieraufwand an Prozeßvariablen auf SIMATIC M7 anbinden.

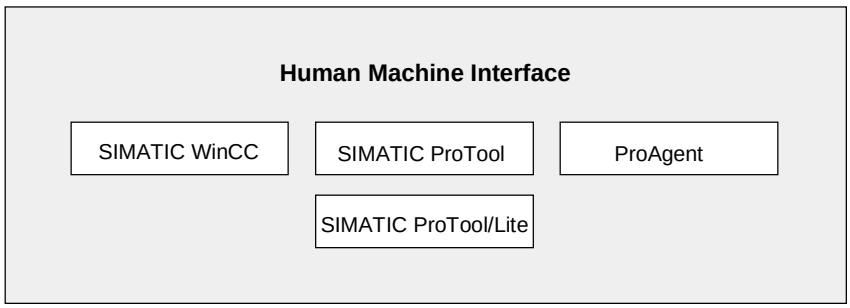
Echtzeitbetriebssystem

- M7-SYS RT enthält das Betriebssystem M7 RMOS 32 und Systemprogramme. Es ist Voraussetzung für den Einsatz der Pakete M7-ProC/C++ und CFC für SIMATIC M7.

1.4.4 Human Machine Interface

Human Machine Interface ist Software speziell zum Bedienen und Beobachten für SIMATIC.

- Das offene Prozeßvisualisierungssystem SIMATIC WinCC ist ein branchen- und technologieunabhängiges Basissystem mit allen wichtigen Funktionen zum Bedienen und Beobachten.
- SIMATIC ProTool und SIMATIC ProTool/Lite sind moderne Tools zur Projektierung der SIMATIC Operator Panels und der SIMATIC C7-Kompaktgeräte.
- ProAgent ermöglicht die gezielte und schnelle Prozeßdiagnose in Anlagen und Maschinen durch die Ermittlung von Informationen zu Fehlerort und Fehlerursache.



2 Installieren und Autorisierung

2.1 Autorisierung

2.1.1 Autorisierung

Für die Nutzung der STEP 7-Programmiersoftware wird eine produktspezifische Autorisierung (Nutzungsberechtigung) benötigt. Die so geschützte Software ist nur benutzbar, wenn auf der Festplatte des betreffenden PG/PC die für das Programm oder Softwarepaket erforderliche Autorisierung erkannt wird.

Jeweils unterschiedliche Autorisierungen sind z. B. erforderlich für STEP 7 und für die Optionssoftware.

2.1.2 Installieren und Deinstallieren der Autorisierung

Autorisierungsdiskette

Für die Autorisierung benötigen Sie die zum Lieferumfang gehörende kopiergeschützte Autorisierungsdiskette. Sie enthält die eigentliche Autorisierung. Das zum Anzeigen, Installieren und Deinstallieren der Autorisierung erforderliche Programm "AuthorsW" befindet sich auf der CD-ROM, die auch STEP 7 V5.1 enthält.

Die Anzahl der möglichen Autorisierungen ist durch einen Autorisierungszähler auf der Autorisierungsdiskette festgelegt. Bei der Autorisierung wird der Zähler um 1 erniedrigt. Steht der Zähler auf dem Wert "Null", ist mit dieser Diskette keine weitere Autorisierung mehr möglich.

Hinweis

Sie erhalten für die Basissoftware STEP 7 eine gelbe Autorisierungsdiskette mit einer zugehörigen Autorisierung.

Für jede Optionssoftware erhalten Sie eine rote Autorisierungsdiskette mit jeweils einer Autorisierung.

Vorsicht

Beachten Sie die Hinweise in der Datei LIESMICH.WRI auf der Autorisierungsdiskette und die Regeln in "Regeln für den Umgang mit Autorisierungen". Bei Nichtbeachtung besteht die Gefahr, daß die Autorisierung unwiderruflich verloren geht.

Sie können die Basissoftware zum kurzen Kennenlernen von Bedienoberfläche und Funktionsumfang auch ohne Autorisierung verwenden. Sinnvolles Arbeiten ist aber nur mit installierter Autorisierung zulässig und möglich. Wenn Sie die Autorisierung nicht installiert haben, werden Sie in regelmäßigen Abständen aufgefordert, die Installation vorzunehmen.

Bei Verlust der Autorisierung ...

Eine Autorisierung kann beispielsweise verloren gehen, wenn ein Festplattendefekt auftritt und Sie keine Möglichkeit mehr haben, die Autorisierung von der defekten Festplatte zu deinstallieren.

Bei Verlust der Autorisierung können Sie auf die Notautorisierung zurückgreifen. Sie befindet sich ebenfalls auf der Autorisierungsdiskette. Die Notautorisierung ermöglicht es Ihnen, die Software zeitlich befristet weiter zu nutzen. In diesem Fall wird beim Start die noch verbleibende Zeit bis zum Ablauf der Gültigkeit angezeigt. Innerhalb dieses Zeitraums sollten Sie sich einen Ersatz für die verlorene Autorisierung beschaffen. Wenden Sie sich dazu bitte an Ihre zuständige SIEMENS-Vertretung.

Hinweis

Die zeitliche Frist der Notautorisierung beginnt mit dem Zeitpunkt der Autorisierungsinstallation abzulaufen, auch dann, wenn STEP 7 nicht gestartet wird. Selbst wenn Sie die Autorisierung zurück auf die Diskette schreiben, wird der Ablauf der zeitlichen Frist nicht gestoppt.

Installation von AuthorsW

Das zum Anzeigen, Installieren und Deinstallieren von Autorisierungen notwendige Programm "AuthorsW" liegt auf der CD-ROM bei, die auch STEP 7 V5.1 enthält. Dieses Programm installieren Sie über ein Setup auf Ihre Festplatte und können es von dort aus für die Autorisierungsoperationen nutzen.

Hinweis

Das Programm AuthorsW befindet sich unter **START > SIMATIC > AuthorsW > AuthorsW**.

Vornehmen der Autorisierung bei der Erstinstallation

Sie sollten die Autorisierung vornehmen, wenn Sie im Rahmen der Erstinstallation von STEP 7 durch eine entsprechende Meldung dazu aufgefordert werden. Gehen Sie dabei wie folgt vor:

1. Legen Sie die Autorisierungsdiskette ein, wenn die entsprechende Aufforderung angezeigt wird.
2. Quittieren Sie anschließend die Aufforderung.
3. Die Berechtigung wird auf ein physikalisches Laufwerk übertragen.

Autorisierung später durchführen

Wenn Sie die STEP 7-Software starten und keine Autorisierung vorhanden ist, so erhalten Sie eine entsprechende Meldung. Um die Autorisierung nachträglich vorzunehmen, gehen Sie wie folgt vor:

1. Legen Sie die Autorisierungsdiskette in das Diskettenlaufwerk, z. B. Laufwerk A:, ein.
2. Rufen Sie das Programm "Authorsw.exe" von der Festplatte auf.
3. Wählen Sie in einem der beiden Listenfelder das Laufwerk, auf welchem sich die Autorisierung befindet, in dem anderen Listenfeld wählen Sie das Ziellaufwerk (z.B. Diskette). Alle auf beiden Laufwerken vorhandenen Autorisierungen werden angezeigt.
4. Markieren Sie die gewünschte Autorisierung.
5. Klicken Sie auf die Schaltfläche "←" oder "→". Die markierte Autorisierung wird auf das gewählte Laufwerk übertragen.
6. Schließen Sie das Dialogfeld.

Hinweis

Die Autorisierung ist unter Windows NT nur dann funktionsfähig, wenn diese Vollzugriff auf das Festplattenlaufwerk “C:” sowie auf das Quelllaufwerk hat.

Hochrüsten der Autorisierung

Zum Hochrüsten von Autorisierungen wählen Sie den Menübefehl “Hochrüsten”. Für diese Funktion benötigen Sie:

- die Autorisierungsdiskette der hochzurüstenden Autorisierung
- das Autorisierungsprogramm "AuthorsW" auf Festplatte
- das neue STEP 7-Upgrade auf Diskette
- die alte Autorisierung auf Diskette oder Festplatte

Bei der Hochrüstprozedur werden die alten Autorisierungen gelöscht und durch neue ersetzt. Die Autorisierungsdiskette darf deshalb zu keinem Zeitpunkt schreibgeschützt sein.

7. Legen Sie die neue Autorisierungsdiskette ein.
8. Rufen Sie das Programm "Authorsw.exe" von der Festplatte auf.
9. Wählen Sie den Menübefehl **Autorisierung > Hochrüsten**. Ein Dialogfeld erscheint.Wählen Sie dort das Upgrade-Programm aus. Danach werden Sie aufgefordert, die Autorisierungsdiskette der alten Autorisierung einzulegen".
10. Legen Sie die gewünschte Autorisierungsdiskette ein. Danach werden Sie gefragt, ob Sie wirklich hochrüsten möchten. Das ist die letzte Möglichkeit, die Aktion abzubrechen. Nach dem Bestätigen des Dialogs darf der Vorgang unter keinen Umständen abgebrochen werden, da sonst die Autorisierung verloren geht.
11. Betätigen Sie die OK-Taste. Danach werden Sie aufgefordert, die Autorisierungsdiskette der neuen Autorisierung einzulegen.

Danach werden alle nötigen Bedingungen überprüft. Nach erfolgreicher Prüfung wird die Hochrüstung mit dem Aktivieren der neuen Autorisierung beendet.

Die neue Autorisierung befindet sich auf dem Laufwerk, auf dem die alte Autorisierung zuletzt war, so daß Sie ggf. die Autorisierung noch von Diskette auf die Festplatte installieren müssen.

Wiederherstellen der Autorisierung

Falls Ihre Autorisierung defekt ist, wenden Sie sich bitte an die Hotline. Eventuell kann dann die Autorisierung mit dem Menübefehl **Autorisierung > Wiederherstellen** gerettet werden.

Deinstallieren der Autorisierung

Wird eine erneute Autorisierung notwendig, z. B. wenn Sie das Laufwerk, auf dem sich die Berechtigung befindet, neu formatieren wollen, so müssen Sie vorher die Berechtigung wieder auf eine Siemens-Autorisierungsdiskette zurückspeichern (deinstallieren). Auf diese Autorisierungsdiskette können Sie auch die Autorisierungen der eingesetzten Options-Pakete zurückspeichern.

Gehen Sie folgendermaßen vor, um die Autorisierung wieder zurück auf die Autorisierungsdiskette zu übertragen:

1. Wenn Sie die Autorisierung auf eine Diskette deinstallieren wollen, legen Sie die Autorisierungsdiskette in das Diskettenlaufwerk, z. B. Laufwerk A:, ein.
2. Rufen Sie das Programm "Authorsw.exe" von der Festplatte auf.
3. Wählen Sie in einem der beiden Listenfelder das Laufwerk, auf welchem sich die Autorisierung befindet, in dem anderen Listenfeld wählen Sie das Ziellaufwerk (z.B. Diskette). Alle auf beiden Laufwerken vorhandenen Autorisierungen werden angezeigt.
4. Markieren Sie die gewünschte Autorisierung.
5. Klicken Sie auf die Schaltfläche "←" oder "→". Die markierte Autorisierung wird auf die Autorisierungsdiskette oder das gewählte Laufwerk übertragen.
6. Schließen Sie das Dialogfeld, wenn Sie keine weiteren Autorisierungen deinstallieren wollen. Sie können diese Diskette erneut zum Autorisieren benutzen.

Es besteht auch die Möglichkeit, Autorisierungen zwischen zwei Festplatten (auch Netzlaufwerken) zu verschieben.

2.1.3 Regeln für den Umgang mit Autorisierungen

Vorsicht

Beachten Sie die Hinweise in diesem Kapitel und in der Datei LIESMICH.TXT auf der Autorisierungsdiskette. Bei Nichtbeachtung besteht die Gefahr, daß die Autorisierung unwiderruflich verloren geht.

Deinstallieren erforderlich

Vor dem Formatieren, Komprimieren oder Restaurieren Ihres Festplattenlaufwerks oder vor dem Installieren eines neuen Betriebssystems müssen Sie zuvor eventuell vorhandene Autorisierungen deinstallieren.

Backup

Enthält ein Backup Ihrer Festplatte Kopien von Autorisierungen, so besteht die Gefahr, daß beim Zurückschreiben der Backup-Daten auf die Festplatte die noch gültig installierten Autorisierungen überschrieben und dadurch zerstört werden.

Um den Verlust der Autorisierung durch Überschreiben eines autorisierten Systems mit einem Backup zu verhindern, müssen Sie

- entweder vor dem Anlegen einer Sicherungskopie alle Autorisierungen entfernen
- oder die Autorisierungen aus der Sicherung ausschließen.

Festplattenoptimierung

Wenn Sie ein Optimierungsprogramm verwenden, das die Möglichkeit anbietet, feste Blöcke zu verschieben, so dürfen Sie diese Option nur verwenden, wenn Sie zuvor die Autorisierungen von der Festplatte auf die Autorisierungsdiskette zurückübertragen haben.

Defekte Sektoren

Mit der Autorisierung entsteht auf dem Ziellaufwerk ein Sonder-Cluster, der manchmal als ”defekt” markiert wird. Versuchen Sie nicht, diesen zu reparieren. Sie könnten dadurch die Autorisierung zerstören.

Schreib- und Kopierschutz

Die Autorisierungsdiskette darf nicht schreibgeschützt sein.

Dateien auf der Autorisierungsdiskette können auf ein anderes Laufwerk (z. B. Festplatte) kopiert und von dort benutzt werden. Eine Autorisierung mit diesen kopierten Dateien ist jedoch nicht möglich; dazu ist die Original-Autorisierungsdiskette erforderlich.

Zulässige Laufwerke

Die Autorisierung kann nur auf dem Festplattenlaufwerk installiert werden. Bei komprimierten Laufwerken (z. B. DBLSPACE) können Sie auf das zugehörige Hostlaufwerk installieren.

Das Autorisierungswerkzeug verhindert die Installation von Autorisierungen auf unzulässigen Laufwerken.

Speicherort

Bei der Autorisierungsinstallation werden die Autorisierungsdateien in dem Schutzverzeichnis "AX NF ZZ mit den Attributen ”System” und ”Versteckt” angelegt.

- Die Attribute dürfen nicht verändert werden.
- Die Dateien dürfen nicht verändert oder gelöscht werden.
- Der Ordner darf nicht verschoben werden. Kopierte Dateien aus dem Ordner (Autorisierungen) werden als fehlerhaft erkannt und sind daher keine zulässigen Autorisierungen.

Die Autorisierung geht sonst unwiederbringlich verloren.

Das Schutzverzeichnis 'AX NF ZZ' wird einmal pro lokalem Laufwerk angelegt. Es enthält alle auf dem Laufwerk installierten Autorisierungen. Es wird bei der Installation der ersten Autorisierung angelegt und beim Entfernen der letzten Autorisierung wieder gelöscht.

Pro Autorisierung werden im Schutzverzeichnis zwei gleichnamige Dateien mit unterschiedlicher Namenserweiterung angelegt. Diese Dateien erhalten den Autorisierungsnamen als Dateinamen.

Anzahl der Autorisierungen

Auf einem Laufwerk können Sie beliebig viele Autorisierungen installieren, solange der erforderliche Speicherplatz zur Verfügung steht; aber nur eine von jeder Version (z. B.: nur eine STEP 7 V4.x und nur eine STEP 7 V5.x). Eine gegenseitige Störung dieser Autorisierungen ist nicht zu befürchten.

Defekte Autorisierung

Defekte Autorisierungen auf einem Festplattenlaufwerk lassen sich mit dem Autorisierungstool AuthorsW nicht mehr entfernen. Sie können sogar die Installation von neuen und gültigen Autorisierungen blockieren. Wenden Sie sich in diesem Fall bitte an Ihre zuständige SIEMENS-Vertretung.

Autorisierungstool

Benutzen Sie die aktuelle Version des Autorisierungstools AuthorsW und möglichst keine älteren Versionen.

Hinweis

Da ab V2.0 des Autorisierungstools nicht alle älteren Autorisierungen erkannt werden können, muß in diesen Fällen mit einer älteren AUTHORS-Version (DOS-Version) < V3.x gearbeitet werden.

2.2 Installieren von STEP 7

2.2.1 Installieren von STEP 7

STEP 7 enthält ein Setup-Programm, das die Installation automatisch durchführt. Eingabeaufforderungen auf dem Bildschirm führen Sie Schritt für Schritt durch den gesamten Installationsvorgang. Es wird mit der unter Windows 95/98/NT bzw. Windows 2000 üblichen Standardprozedur zur Installation von Software aufgerufen.

Die wesentlichen Phasen der Installation sind:

- das Kopieren der Daten auf Ihr Erstellsystem,
- das Einrichten der Treiber für EPROM und Kommunikation,
- das Eintragen der Ident-Nr.
- die Autorisierung (falls gewünscht).

Hinweis

Siemens Programmiergeräte (wie zum Beispiel das PG 740) werden mit installierbarer STEP 7-Software auf der Festplatte ausgeliefert.

Installationsvoraussetzungen

- Betriebssystem
  Microsoft Windows 95, Windows 98, Windows 2000 oder Windows NT.
- Basishardware
  PC oder Programmiergerät mit
  - Prozessor 80486 oder höher, (bei Windows NT/2000 Pentium-Prozessor) und
  - RAM-Speicherausbau: mindestens 32 MB, empfohlen 64 MB.
  - Farbmonitor, Tastatur und Maus, die von Microsoft Windows unterstützt werden.

Ein Programmiergerät (PG) ist ein Personal Computer in spezieller industrietauglicher und kompakter Ausführung. Es ist komplett ausgestattet für die Programmierung der SIMATIC-Automatisierungssysteme.

- Speicherkapazität
  Erforderlicher Speicherplatz auf der Festplatte siehe “Readme”.
- MPI-Schnittstelle (optional)
  Die MPI-Schnittstelle zwischen Erstellsystem (Programmiergerät oder PC) und Zielsystem ist nur erforderlich, wenn Sie unter STEP 7 über MPI mit dem Zielsystem kommunizieren wollen.
  Dazu verwenden Sie entweder
  - ein PC/MPI-Kabel, das an die Kommunikationsschnittstelle Ihres Geräts angeschlossen wird oder

- eine MPI-Baugruppe, die in Ihrem Gerät installiert wird.

Bei Programmiergeräten ist die MPI-Schnittstelle bereits eingebaut.

- Externer Prommer (optional)
  Ein externer Prommer ist bei Einsatz eines PCs nur dann erforderlich, wenn Sie EPROMs schießen wollen.

Hinweise

Beachten Sie auch die Hinweise zum Installieren von STEP 7 in der Datei LIESMICH.WRI und die "Liste der Verträglichkeit von SIMATIC-Softwarepaketen mit den Versionen des STEP 7-Basispaketes".

Die Liesmich-Datei finden Sie in der Startleiste unter **Start > Simatic > Produkt-Hinweise**.

Die Verträglichkeitsliste finden Sie in der Startleiste unter **Start > Simatic > Dokumentation**.

2.2.2 Vorgehen beim Installieren

Vorbereitungen

Bevor Sie mit der Installation beginnen können, muß Windows 95/98/NT/2000 gestartet sein.

- Sie benötigen keine externen Datenträger, wenn sich die installierbare STEP 7-Software bereits auf der Festplatte des PG befindet.
- Um von Diskette zu installieren, legen Sie Diskette 1 in das Diskettenlaufwerk Ihres Programmiergeräts/PC ein.
- Um von CD-ROM zu installieren, legen Sie die CD-ROM in das CD-ROM-Laufwerk ihres PC.

Starten des Installationsprogramms

Gehen Sie zur Installation wie folgt vor:

1. Legen Sie den Datenträger (Diskette 1) oder die CD-ROM ein und starten Sie das Setup durch Doppelklick auf die Datei ”setup.exe”.
2. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Das Programm führt Sie schrittweise durch den Installationsprozeß. Sie können jeweils zum nachfolgenden oder vorhergehenden Schritt weiterschalten.

Während des Installationsvorgangs werden Ihnen in Dialogfeldern Fragen angezeigt oder Optionen zur Auswahl angeboten. Lesen Sie bitte die folgenden Hinweise, um die Dialoge schneller und leichter beantworten zu können.

Falls schon eine STEP 7-Version installiert ist ...

Wenn das Installationsprogramm feststellt, daß sich bereits eine STEP 7-Installation auf dem Erstellsystem befindet, wird eine entsprechende Meldung angezeigt, und Sie haben folgenden Wahlmöglichkeiten:

- Installation abbrechen (um danach alte STEP 7-Version unter Windows zu deinstallieren und anschließend die Installation erneut zu starten) oder
- Installation fortsetzen und damit alte Version durch neue Version überschreiben.

Hinsichtlich einer sauberen Softwarepflege sollten Sie vor einer Installation eine eventuell vorhandene ältere Version deinstallieren. Das einfache Überschreiben einer älteren Version hat außerdem den Nachteil, daß bei einem anschließenden Deinstallieren die eventuell noch vorhandenen Teile aus einer älteren Installation nicht entfernt werden.

Zum Installationsumfang

Zur Festlegung des Installationsumfangs haben Sie drei Auswahlmöglichkeiten:

- Maximalkonfiguration: alle Sprachen der Benutzungsoberfläche, alle Applikationen und alle Beispiele. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Minimalkonfiguration: nur eine Sprache, keine Beispiele. Den dazu benötigten Speicherplatz entnehmen Sie bitte der aktuellen Produktinformation.
- Benutzerdefiniert: Sie können den Installationsumfang wählen hinsichtlich Programme, Datenbasis, Beispiele und Kommunikation.

Zur Ident-Nummer

Sie werden bei der Installation nach einer Ident-Nummer gefragt. Tragen Sie die Ident-Nummer ein. Sie finden diese auf dem Software-Produktschein oder auf der zugehörigen Autorisierungsdiskette.

Zur Autorisierung

Bei der Installation wird überprüft, ob eine Autorisierung auf der Festplatte vorhanden ist. Wird keine Autorisierung erkannt, so erscheint ein Hinweis, daß die Software nur mit Autorisierung benutzt werden kann. Wenn Sie es wünschen, können Sie gleich die Autorisierung vornehmen oder aber die Installation fortsetzen und die Autorisierung zu einem späteren Zeitpunkt nachholen. Im erstgenannten Fall legen Sie die Autorisierungsdiskette ein, wenn Sie dazu aufgefordert werden.

Zur PG/PC-Schnittstelleneinstellung

Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Lesen Sie dazu "Einstellen der PG/PC-Schnittstelle".

Zur Memory Card-Parametrierung

Während des Installationsvorgangs wird ein Dialog zur Memory Card-Parametrierung angezeigt.

- Wenn Sie keine Memory Cards einsetzen, benötigen Sie keinen EPROM-Treiber. Wählen Sie die Option ”kein EPROM-Treiber”.
- Ansonsten wählen Sie den Eintrag, der zu Ihrem PG gehört.
- Wenn Sie einen PC einsetzen, so können Sie einen Treiber für einen externen Prommer auswählen. Dabei müssen Sie zusätzlich die Schnittstelle angeben, an der der Prommer angeschlossen ist (z. B. LPT1).

Sie können die eingestellten Parameter nach der Installation ändern, wenn Sie das Programm ”Memory Card parametrieren” in der STEP 7-Programmgruppe oder der Systemsteuerung aufrufen.

Zum Flash-File-System

Im Dialog zur Memory Card-Parametrierung können Sie angeben, ob ein Flash-File-System installiert werden soll.

Das Flash-File-System wird z. B. benötigt, wenn Sie bei SIMATIC M7 einzelne Dateien auf eine EPROM-Memory Card schreiben oder dort löschen wollen, ohne den restlichen Inhalt der Memory Card zu verändern.

Wenn Sie ein geeignetes Programmiergerät (PG 720/740/760) oder einen geeigneten externen Prommer einsetzen und diese Funktionalität nutzen wollen, so wählen Sie die Installation des Flash-File-Systems.

Fehler während der Installation

Folgende Fehler führen zum Abbruch der Installation:

- Wenn sofort nach dem Start von Setup ein Initialisierungsfehler auftritt, so wurde höchstwahrscheinlich *setup* nicht unter Windows gestartet.
- Speicherplatz reicht nicht aus: Sie benötigen abhängig vom Installationsumfang ca. 100 MB freien Speicherplatz auf Ihrer Festplatte für die Basissoftware.
- Defekte Diskette oder CD: Wenn Sie feststellen, daß eine Diskette defekt ist, wenden Sie sich bitte an Ihre SIEMENS-Vertretung.
- Bedienungsfehler: Beginnen Sie die Installation erneut und beachten Sie die Anweisungen sorgfältig.

Zum Abschluß der Installation ...

Wenn die Installation erfolgreich war, wird dies durch eine entsprechende Meldung auf dem Bildschirm angezeigt.

Falls bei der Installation Systemdateien aktualisiert wurden, werden Sie aufgefordert, Windows neu zu starten. Nach dem Neustart können Sie die Oberfläche von STEP 7, den SIMATIC Manager starten.

Nach einer erfolgreichen Installation ist eine Programmgruppe für STEP 7 eingerichtet.

2.2.3 Einstellen der PG/PC-Schnittstelle

Mit den hier gemachten Einstellungen legen Sie die Kommunikation zwischen PG/PC und dem Automatisierungssystem fest. Während des Installationsvorgangs wird ein Dialog zur Einstellung der PG/PC-Schnittstelle angezeigt. Sie können sich den Dialog auch nach der Installation anzeigen lassen, indem Sie das Programm ”PG/PC-Schnittstelle einstellen” aufrufen. Dadurch ist es möglich, die Schnittstellenparameter auch später unabhängig von einer Installation zu ändern.

Prinzipielles Vorgehen

Zum Betrieb einer Schnittstelle sind erforderlich:

- Einstellungen im Betriebssystem
- eine geeignete Schnittstellenparametrierung

Wenn Sie ein PG über eine MPI-Verbindung einsetzen, so sind keine weiteren betriebssystemspezifischen Anpassungen notwendig.

Wenn Sie einen PC mit MPI-Karte oder Kommunikationsprozessoren (CP) einsetzen, so sollten Sie über die ”Systemsteuerung” von Windows die Interrupt- und Adreßbelegung prüfen, um sicherzustellen, daß es zu keinen Interrupt-Konflikten oder Adreßbereichsüberschneidungen kommt.

Um die Parametrierung der PG/PC-Schnittstelle zu vereinfachen, werden Ihnen in Dialogfeldern vordefinierte Sätze von Grundparametern (Schnittstellenparametrierungen) zur Auswahl angeboten.

Parametrieren der PG/PC-Schnittstelle

Gehen Sie folgendermaßen vor (ausführliche Beschreibung in der Online-Hilfe):

1. Doppelklicken Sie in der ”Systemsteuerung” auf ”PG/PC-Schnittstelle einstellen”.
2. Stellen Sie den ”Zugangspunkt der Applikation” auf ”S7ONLINE”.
3. Wählen Sie in der Liste "Benutzte Schnittstellenparametrierung" die gewünschte Schnittstellenparametrierung aus. Wird die von Ihnen gewünschte Schnittstellenparametrierung nicht angezeigt, müssen Sie zunächst über die Schaltfläche ”Auswählen” eine Baugruppe bzw. ein Protokoll installieren. Die Schnittstellenparametrierung wird dann automatisch erstellt.

- Wenn Sie eine Schnittstelle mit automatischer Erkennung der Busparameter wählen (z. B. MPI-ISA-Card (Auto)), dann können Sie das PG bzw. den PC an MPI bzw. PROFIBUS anschließen, ohne Busparameter einstellen zu müssen. Bei Übertragungsgeschwindigkeiten kleiner 187,5 kBit/s können allerdings Wartezeiten bis zu einer Minute entstehen.
  Voraussetzung für die automatsche Erkennung: Am Bus sind Master angeschlossen, die zyklisch Busparameter verteilen. Alle neuen MPI-Komponenten tun das; bei PROFIBUS-Subnetzen darf das zyklische Verteilen der Busparameter nicht ausgeschaltet sein (voreingestellte PROFIBUS-Netzeinstellung).
- Wenn Sie eine Schnittstelle ohne automatische Erkennung der Busparameter wählen, dann lassen Sie sich die Eigenschaften anzeigen und passen sie an das Subnetz an.

Änderungen sind auch dann erforderlich, wenn sich Konflikte mit anderen Einstellungen ergeben (z. B. Interrupt- oder Adreßbelegungen). Nehmen Sie in diesem Fall die entsprechenden Änderungen mit der Hardwareerkennung und der Systemsteuerung von Windows vor (siehe unten).

Vorsicht

Eine eventuell angezeigte Schnittstellenparametrierung ”TCP/IP” **nicht** entfernen! Dies könnte den Ablauf von anderen Anwendungen beeinträchtigen.

Prüfen der Interrupt- und Adreßbelegung

Wenn Sie einen PC mit MPI-Karte einsetzen, so sollten Sie in jedem Fall prüfen, ob der voreingestellte Interrupt und der voreingestellte Adreßbereich frei sind und ggf. einen freien Interrupt und/oder Adreßbereich wählen.

Windows95/98

Sie können sich die aktuellen Belegungen unter Windows 95/98 wie folgt anzeigen lassen:

1. Öffnen Sie in der ”Systemsteuerung” das ”System” und wählen Sie im angezeigten Dialogfeld das Register ”Geräte-Manager”.
2. Markieren Sie in der angezeigten Liste den Eintrag ”Computer” und betätigen Sie die Schaltfläche ”Eigenschaften”.
3. In einem weiteren Dialogfeld können Sie sich die Liste der belegten Interrupts (IRQ) oder die Liste der belegten Adreßbereiche (Ein-/Ausgabe (E/A)) durch Markieren des entsprechenden Optionsfeldes anzeigen lassen.

Windows NT

Unter Windows NT können Sie sich

- die Ressourceneinstellungen unter **START > Programme > Verwaltung (Allgemein) > Windows NT-Diagnose > Ressourcen** anzeigen lassen.
- die Ressourcen unter **PG/PC-Schnittstelle > Installieren > Ressourcen** verändern.

Windows 2000

Unter Windows 2000 können Sie sich die Ressourcen anzeigen lassen

- unter **Systemsteuerung > Verwaltung > Computerverwaltung > System > Systeminformationen > Hardwareressourcen**.

Unterschiede zwischen Windows 9x und Windows NT/2000

Die Vergabe von Interrupts, Adreßbereichen und anderen Ressourcen muß unter Windows NT/2000 in einem eigenen Dialog vorgenommen werden (eine ausführliche Beschreibung finden Sie in der Online-Hilfe).

2.3 Deinstallieren von STEP 7

2.3.1 Deinstallieren von STEP 7

Benutzen Sie das unter Windows übliche Verfahren zur Deinstallation:

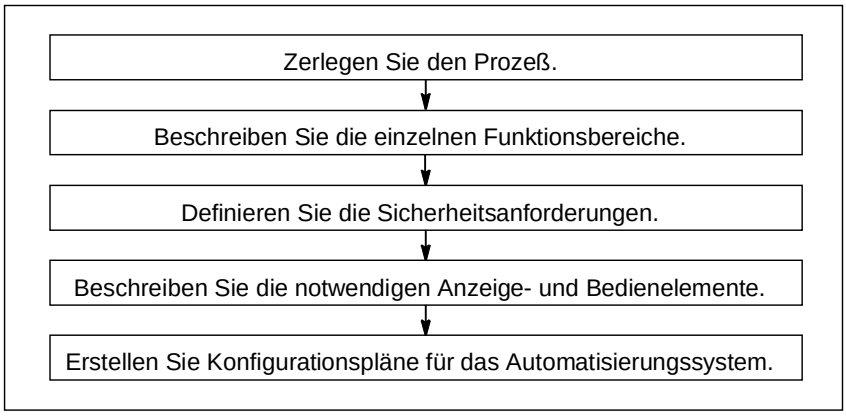
1. Starten Sie unter Windows den Dialog zu Installation von Software durch Doppelklick auf das Symbol ”Software” in ”Systemsteuerung”.
2. Markieren Sie den STEP 7-Eintrag in der angezeigten Liste der installierten Software. Betätigen Sie dann die Schaltfläche zum ”Entfernen” der Software.
3. Falls Dialogfelder ”Freigegebene Datei entfernen” erscheinen, so klicken Sie im Zweifelsfall auf die Schaltfläche ”Nein”.

3 Konzeption der Automatisierungslösung

3.1 Prinzipielle Vorgehensweise zur Konzeption einer Automatisierungslösung

Dieses Kapitel enthält Informationen zu den grundlegenden Aufgaben beim Planen einer Automatisierungslösung für ein Automatisierungssystem (AS). Anhand eines Beispiels zur Automatisierung eines industriellen Mischprozesses wird beschrieben, wie Sie Schritt für Schritt dabei vorgehen können.

Es gibt viele Methoden, eine Automatisierungslösung vorzubereiten. Die grundsätzliche Vorgehensweise, die Sie für jedes beliebige Projekt verwenden können, ist im nachfolgenden Bild dargestellt.

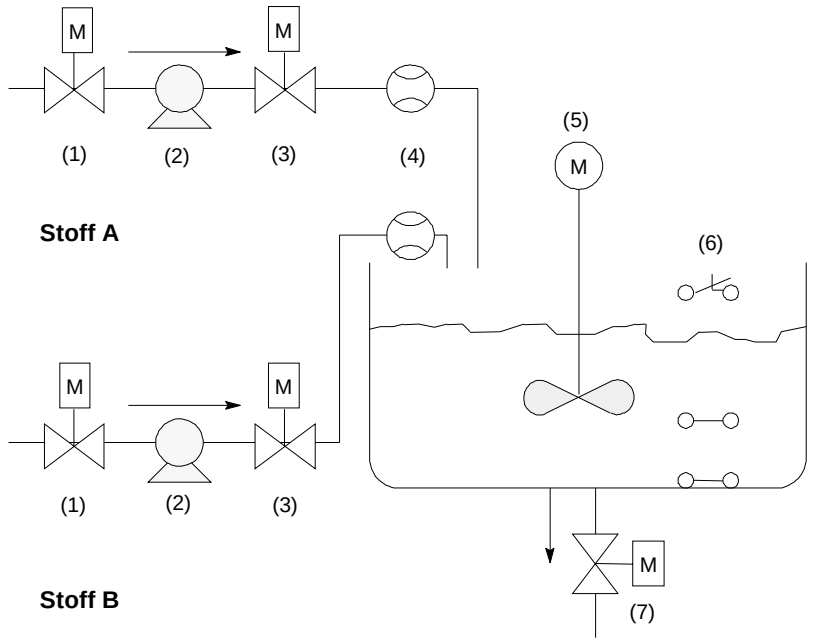

Zerlegen Sie den Prozeß.
Beschreiben Sie die einzelnen Funktionsbereiche.
Definieren Sie die Sicherheitsanforderungen.
Beschreiben Sie die notwendigen Anzeige- und Bedienelemente.
Erstellen Sie Konfigurationspläne für das Automatisierungssystem.

3.2 Zerlegen des Prozesses in Aufgaben und Bereiche

Ein Automatisierungsprozeß besteht aus einzelnen Aufgaben. Selbst der komplizierteste Prozeß kann definiert werden, wenn Sie zusammenhängende Aufgabenbereiche innerhalb des Prozesses bestimmen und diese in kleinere Teilaufgaben zerlegen.

Im folgenden Beispiel wird anhand eines industriellen Mischprozesses gezeigt, wie Sie einen Prozeß in Funktionsbereiche und einzelne Aufgaben strukturieren können:

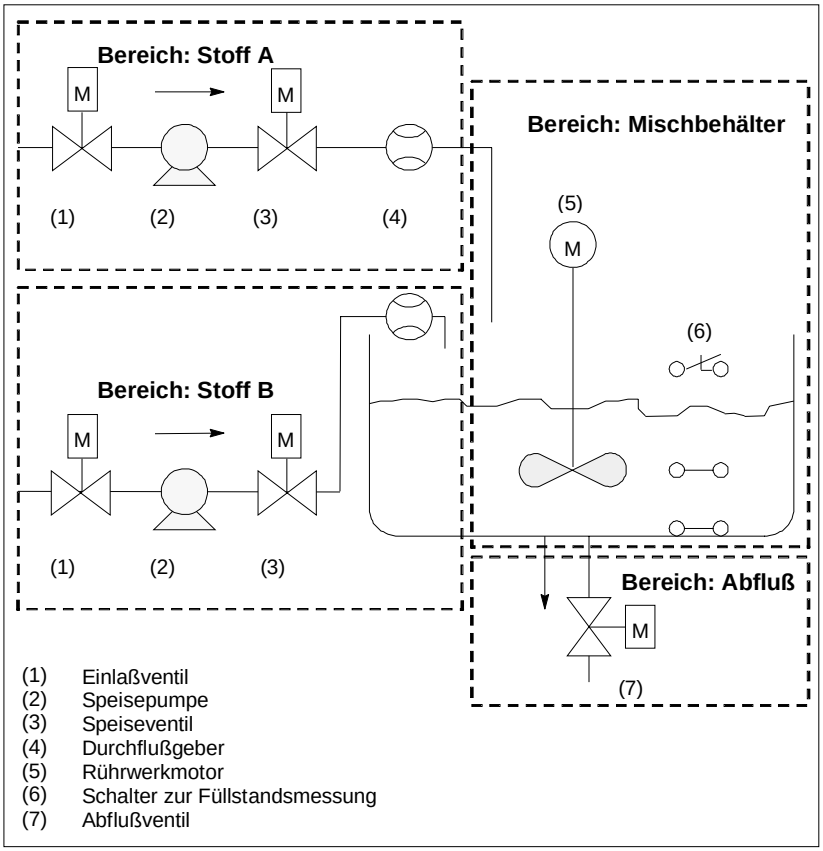
Beispiel: "Industrieller Mischprozeß"



(1) Einlaßventil
(2) Speisepumpe
(3) Speiseventil
(4) Durchflußgeber
(5) Rührwerkmotor
(6) Schalter zur Füllstandsmessung
(7) Magnetschalter Abflußventil

Bestimmen der Bereiche eines Prozesses

Nachdem Sie den Prozeß, der gesteuert werden soll, definiert haben, unterteilen Sie das Projekt in zusammengehörige Gruppen bzw. Bereiche:



Da jeder Bereich in kleinere Aufgaben unterteilt wird, sind die Aufgaben zur Steuerung dieses Prozeßabschnitts nicht sehr kompliziert.

Im Beispiel des industriellen Mischprozesses können Sie vier Bereiche bestimmen (siehe nachfolgende Tabelle). In diesem Beispiel enthält der Bereich für Stoff A die gleichen Geräte wie der Bereich für Stoff B.

Funktionsbereich	**Zugehörige Geräte**
Stoff A	Speisepumpe für Stoff A Einlaßventil für Stoff A Speiseventil für Stoff A Durchflußgeber für Stoff A
Stoff B	Speisepumpe für Stoff B Einlaßventil für Stoff B Speiseventil für Stoff B Durchflußgeber für Stoff B
Mischbehälter	Rührwerkmotor Schalter zur Füllstandsmessung
Abfluß	Abflußventil

3.3 Beschreiben der einzelnen Funktionsbereiche

Wenn Sie jeden Bereich und jede Aufgabe in Ihrem Prozeß beschreiben, dann definieren Sie nicht nur die Funktionsweise jedes Bereichs, sondern auch die verschiedenen Elemente, die diesen Bereich steuern. Diese umfassen:

- Elektrische, mechanische und logische Eingänge und Ausgänge für jede Aufgabe
- Verriegelungen und Abhängigkeiten zwischen den einzelnen Aufgaben

In dem Beispiel des industriellen Mischprozesses werden Pumpen, Motoren und Ventile eingesetzt. Diese müssen genau beschrieben werden, um die Betriebsmerkmale und die Art der Verriegelungen festzulegen, die während des Betriebs erforderlich sind. In den nachfolgenden Tabellen finden Sie Beispiele für die Beschreibung der Geräte, die in dem industriellen Mischprozeß eingesetzt werden. Diese Beschreibung können Sie auch für die Beschaffung der benötigten Geräte verwenden.

Stoff A/B: Speisepumpenmotoren
Die Speisepumpen fördern die Stoffe A und B in den Mischbehälter. • Durchflußleistung: 400 l pro Minute • Leistung: 100 KW bei 1200 UPM
Die Pumpen werden von einem Bedienpult gesteuert (Starten/Stoppen), das sich in der Nähe des Mischbehälters befindet. Die Anzahl der Starts wird zu Wartungszwecken gezählt. Der Zähler und die Anzeige können mit einem gemeinsamen Taster zurückgesetzt werden.
Es gelten die folgenden Freigabebedingungen: • Der Mischbehälter ist nicht voll. • Das Abflußventil des Mischbehälters ist geschlossen. • NOT-AUS ist nicht aktiviert.
Es gilt die folgende Abschaltbedingung: • Durchflußgeber meldet 7 s nach Start des Pumpenmotors keinen Durchfluß. • Durchflußgeber meldet während des Laufs keinen Durchfluß mehr.

Stoff A/B: Einlaß- und Speiseventile
Die Einlaß- und Speiseventile für die Stoffe A und B ermöglichen/verhindern den Zufluß der Stoffe in den Mischbehälter. Die Ventile haben einen Magnetschalter mit Federrückholeinrichtung. • Wenn der Magnetschalter aktiviert ist, ist das Ventil geöffnet. • Wenn der Magnetschalter deaktiviert ist, ist das Ventil geschlossen.
Die Einlaß- und Speiseventile werden vom Anwenderprogramm gesteuert.
Es gilt die folgende Freigabebedingung: • Der Speisepumpenmotor ist mindestens 1 Sekunde angesteuert.
Es gilt die folgende Abschaltbedingung: • Der Durchflußgeber meldet keinen Durchfluß.

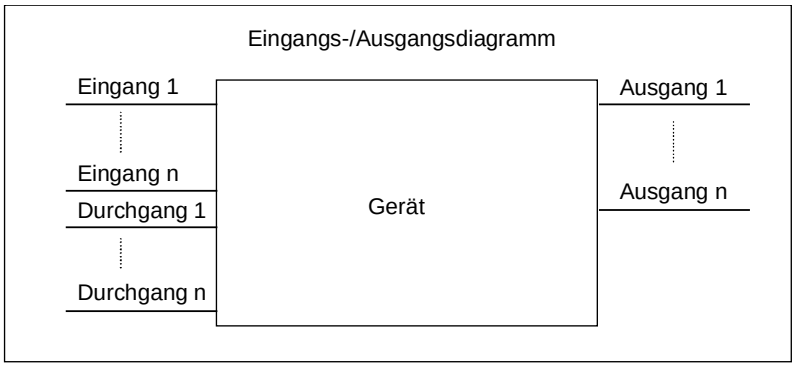
Rührwerkmotor
Mit dem Rührwerkmotor werden die Stoffe A und B im Mischbehälter gemischt. • Leistung: 100 KW bei 1200 UPM
Der Rührwerkmotor wird von einem Bedienpult gesteuert (Starten/Stoppen), das sich in der Nähe des Mischbehälters befindet. Die Anzahl der Starts wird zu Wartungszwecken gezählt. Der Zähler und die Anzeige können mit einem gemeinsamen Taster zurückgesetzt werden.
Es gelten die folgenden Freigabebedingungen: • Der Füllstandsmesser zeigt nicht ”Behälter unter Minimum”. • Das Abflußventil des Mischbehälters ist geschlossen. • NOT-AUS ist nicht aktiviert.
Es gilt die folgende Abschaltbedingung: • Drehzahlgeber meldet nicht spätestens 10 s nach Start des Motors, daß die Nenndrehzahl erreicht ist.

Abflußventil
Durch Fallspeisung gelangt das Gemisch durch das Abflußventil zum nächsten Prozeßschritt. Das Ventil hat einen Magnetschalter mit Federrückholeinrichtung. • Wenn der Magnetschalter aktiviert ist, ist das Abflußventil geöffnet. • Wenn der Magnetschalter deaktiviert ist, ist das Abflußventil geschlossen.
Das Abflußventil wird vom Bedienpult gesteuert (Öffnen/Schließen).
Das Abflußventil kann unter folgenden Bedingungen geöffnet werden: • Der Rührwerkmotor ist abgeschaltet. • Der Füllstandsmesser zeigt nicht ”Behälter ist leer”. • NOT-AUS ist nicht aktiviert.
Es gilt folgende Abschaltbedingung: • Die Füllstandsmessung ergibt “Behälter leer”.

Schalter zur Füllstandsmessung
Die Schalter im Mischbehälter informieren über den Füllstand im Behälter und werden für die Verriegelung der Speisepumpen und des Rührwerkmotors verwendet.

3.4 Auflisten von Ein-, Aus- und Durchgängen

Nachdem Sie jedes Gerät, das gesteuert werden soll, physikalisch beschrieben haben, zeichnen Sie für jedes Gerät oder jeden Aufgabenbereich ein Eingangs-/Ausgangsdiagramm.



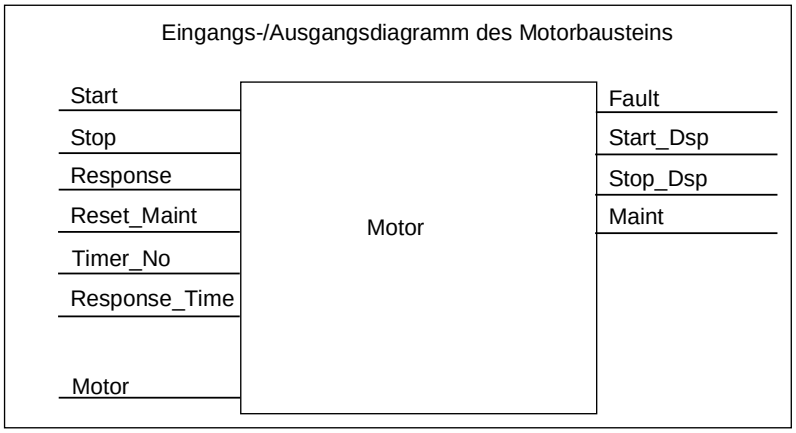
Diese Diagramme entsprechen den Codebausteinen, die zu programmieren sind.

3.5 Erstellen eines Eingangs-, Ausgangs-Diagramms für die Motoren

In dem Beispiel für den industriellen Mischprozeß werden zwei Speisepumpen und ein Rührwerk eingesetzt. Die Ansteuerung der jeweiligen Motoren erfolgt durch je einen “Motorbaustein”, der für alle drei Geräte gleich ist. Dieser Baustein benötigt sechs Eingänge: Zwei Eingänge zum Starten bzw. Stoppen des Motors, einen Eingang zum Rücksetzen der Wartungsanzeige, einen Eingang für die Motorrückmeldung (Motor läuft / läuft nicht), einen Eingang für die Zeitdauer, in der die Rückmeldung erfolgen muß, und einen Eingang für die Nummer des Timers, der zur Zeitmessung verwendet werden soll.

Außerdem benötigt der Codebaustein vier Ausgänge: Zwei Ausgänge zur Anzeige des Betriebszustandes des Motors, einen Ausgang zur Anzeige von Fehlern und einen Ausgang, der anzeigt, daß die Wartung des Motors fällig ist.

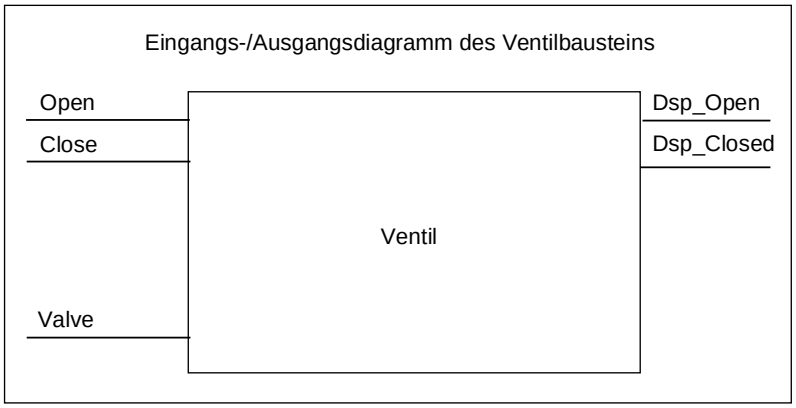
Weiterhin wird ein Durchgang benötigt: Er dient zur Ansteuerung des Motors, wird aber gleichzeitig im Programm des “Motorbausteines” bearbeitet bzw. verändert.



3.6 Erstellen eines Eingangs-/ Ausgangs-Diagramms für die Ventile

Die Ansteuerung des jeweiligen Ventils erfolgt über einen “Ventilbaustein”, der für alle verwendeten Ventile gleich ist. Der Codebaustein hat zwei Eingänge: Einen Eingang zum Öffnen und einen zum Schließen des Ventils. Außerdem sind zwei Ausgänge vorhanden: Ein Ausgang zeigt an, daß das Ventil geöffnet ist, der andere, daß es geschlossen ist.

Der Baustein hat einen Durchgang: Er dient zur Ansteuerung des Ventils, wird aber gleichzeitig im Programm des “Ventilbausteines” bearbeitet bzw. verändert.



3.7 Festlegen der Sicherheitsanforderungen

Entscheiden Sie - im Rahmen der gesetzlichen Vorschriften und der Verfahrensanweisungen Ihres Unternehmens - welche Elemente notwendig sind, um die Sicherheit des Prozesses zu gewährleisten. Beschreiben Sie auch, wie diese Sicherheitselemente die Bereiche Ihres Prozesses beeinflussen.

Definieren der Sicherheitsanforderungen

Bestimmen Sie die Geräte, die aus Sicherheitsgründen festverdrahtete Schaltungen benötigen. Per Definition arbeiten diese Sicherheitsstromkreise unabhängig vom Automatisierungssystem (obwohl der Sicherheitsstromkreis in der Regel eine E/A-Schnittstelle zur Koordination mit dem Anwenderprogramm zur Verfügung stellt). Üblicherweise konfigurieren Sie eine Matrix, um jeden Aktor mit einem eigenen NOT-AUS-Bereich zu verbinden. Diese Matrix ist die Grundlage für die Schaltpläne der Sicherheitsstromkreise.

Zum Entwerfen von Schutzvorrichtungen gehen Sie wie folgt vor:

- Legen Sie die logischen und mechanischen/elektrischen Verriegelungen zwischen den einzelnen Automatisierungsaufgaben fest.
- Entwerfen Sie Schaltungen, um im Notfall die am Prozeß beteiligten Geräte manuell bedienen zu können.
- Bestimmen Sie weitere Sicherheitsanforderungen für einen sicheren Ablauf des Prozesses.

Erstellen eines Sicherheitsstromkreises

In dem Beispiel für den industriellen Mischprozeß verwenden wir für den Sicherheitsstromkreis folgende Schaltung:

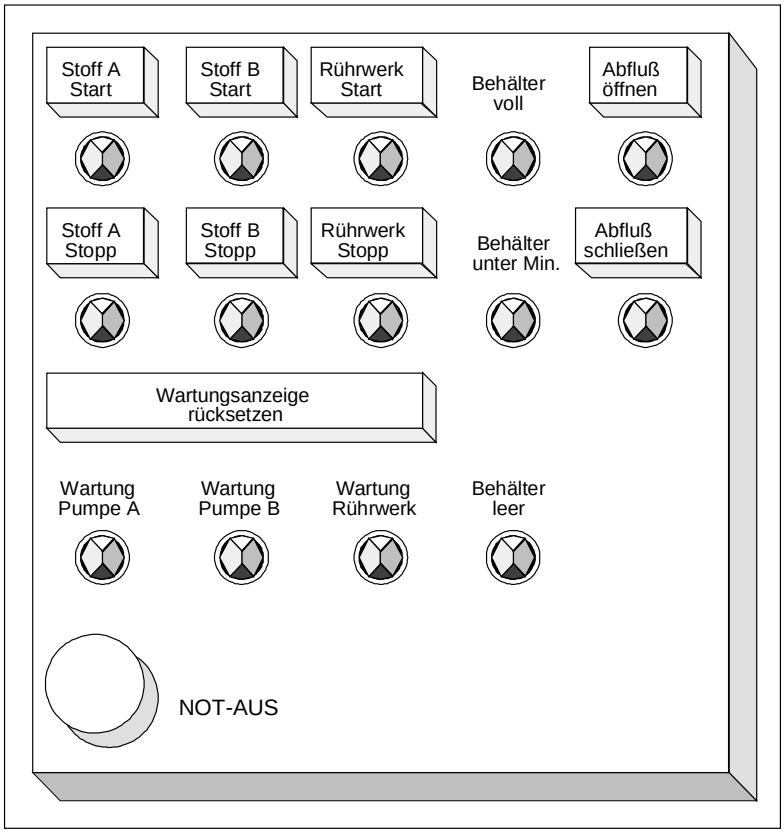
- Einen NOT-AUS-Schalter, der unabhängig vom Automatisierungssystem (AS) die folgenden Geräte ausschaltet:
  - Speisepumpe für Stoff A
  - Speisepumpe für Stoff B
  - Rührwerkmotor
  - Ventile
- Der NOT-AUS-Schalter befindet sich am Bedienpult.
- Ein Eingang der Steuerung erfaßt den Zustand des NOT-AUS-Schalters.

3.8 Beschreiben der notwendigen Anzeige- und Bedienelemente

Jeder Prozeß benötigt ein Bedien- und Beobachtungssystem, das menschliches Eingreifen in den Prozeß ermöglicht. Als Teil der Entwurfsbeschreibung wird auch die Ausführung eines Bedienpultes beschrieben.

Definieren eines Bedienpultes

Beim industriellen Mischprozeß in unserem Beispiel wird jedes Gerät über einen Taster, der sich auf dem Bedienpult befindet, gestartet bzw. gestoppt. Dieses Bedienpult ist mit Anzeigen versehen, die über den Betriebszustand informieren (siehe nachfolgendes Bild).



Hier befinden sich auch die Anzeigelampen für die Geräte, die nach einer bestimmten Anzahl von Starts gewartet werden müssen, und der NOT-AUS-Schalter, mit dem der Prozeß sofort angehalten werden kann. Auf dem Bedienpult befindet sich außerdem ein Rücksetztaster für die Wartungsanzeige der drei Motoren. Mit ihm schalten Sie die Wartungsanzeigelampen für diejenigen Motoren aus, bei denen eine Wartung fällig ist, und setzen die zugehörigen Zählerstände für das Wartungsintervall auf 0.

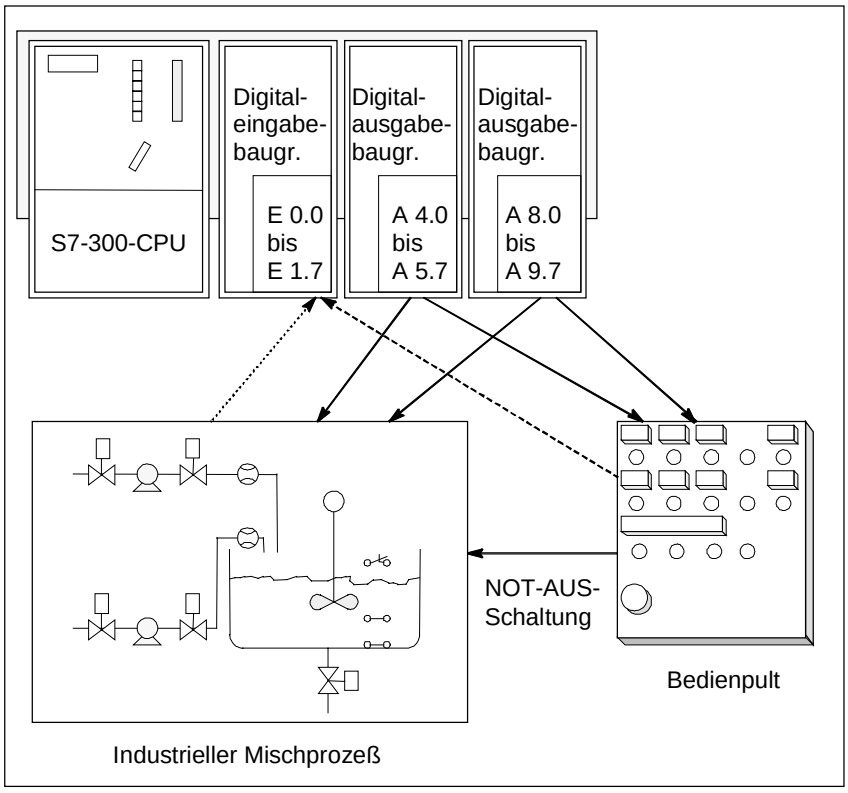
3.9 Erstellen des Konfigurationsplans

Nachdem Sie die Entwurfsanforderungen dokumentiert haben, bestimmen Sie die Steuergeräte, die für dieses Projekt erforderlich sind.

Mit der Entscheidung, welche Baugruppen Sie einsetzen wollen, legen Sie den Aufbau des Automatisierungssystems fest. Erstellen Sie einen Konfigurationsplan, in dem Sie folgende Punkte festlegen:

- Art der CPU
- Anzahl und Art der Signalbaugruppen
- Konfiguration der physikalischen Ein- und Ausgänge

Nachfolgendes Bild zeigt das Beispiel einer S7-Konfiguration für den industriellen Mischprozeß.



4 Grundlagen zum Entwerfen einer Programmstruktur

4.1 Programme in einer CPU

In einer CPU laufen grundsätzlich zwei verschiedene Programme ab:

- das Betriebssystem und
- das Anwenderprogramm.

Betriebssystem

Das Betriebssystem ist bereits in jeder CPU enthalten und organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind. Zu seinen Aufgaben gehören:

- das Abwickeln von Neustart und Wiederanlauf
- das Aktualisieren des Prozeßabbildes der Eingänge und die Ausgabe des Prozessabbildes der Ausgänge
- das Aufrufen des Anwenderprogramms
- das Erfassen von Alarmen und das Aufrufen der Alarm-OBs
- das Erkennen und Behandeln von Fehlern
- das Verwalten von Speicherbereichen
- das Kommunizieren mit Programmiergeräten und anderen Kommunikationspartnern.

Durch Ändern der Betriebssystemparameter (Voreinstellungen des Betriebssystems) können Sie das Verhalten der CPU in bestimmten Bereichen beeinflussen.

Anwenderprogramm

Das Anwenderprogramm müssen Sie erstellen und in die CPU laden. Es enthält alle Funktionen, die zur Bearbeitung Ihrer spezifischen Automatisierungsaufgabe erforderlich sind. Zu den Aufgaben des Anwenderprogramms gehören:

- das Festlegen der Voraussetzungen für den Neustart und den Wiederanlauf der CPU (z. B. Signale mit einem bestimmten Wert vorbesetzen)
- das Bearbeiten von Prozeßdaten (z. B. Binärsignale verknüpfen, Analogwerte einlesen und auswerten, Binärsignale für die Ausgabe festlegen, Analogwerte ausgeben)
- das Reagieren auf Alarme
- das Bearbeiten von Störungen im normalen Programmablauf.

4.2 Bausteine im Anwenderprogramm

4.2.1 Bausteine im Anwenderprogramm

Die Programmiersoftware STEP 7 bietet Ihnen die Möglichkeit, das Anwenderprogramm zu strukturieren, d. h. in einzelne, in sich geschlossene Programmabschnitte aufzuteilen. Daraus ergeben sich die folgenden Vorteile:

- umfangreiche Programme lassen sich übersichtlich programmieren
- einzelne Programmteile können standardisiert werden
- die Programmorganisation wird vereinfacht
- Änderungen des Programms lassen sich leichter durchführen
- der Programmtest wird vereinfacht, weil er abschnittsweise erfolgen kann
- die Inbetriebnahme wird erleichtert.

Am Beispiel des industriellen Mischprozesses haben Sie gesehen, wie ein Automatisierungsprozeß sinnvoll in einzelne Aufgaben zerlegt werden kann. Die Programmabschnitte eines strukturierten Anwenderprogramms entsprechen diesen einzelnen Aufgaben, sie werden als Bausteine eines Programms bezeichnet.

Bausteinarten

Es gibt verschiedene Arten von Bausteinen, die Sie innerhalb eines S7-Anwenderprogramms verwenden können:

Baustein	Kurzbeschreibung der Funktion	Näheres siehe
Organisationsbausteine (OB)	OBs legen die Struktur des Anwender-programms fest.	"Organisationsbausteine und Programmstruktur"
Systemfunktionsbausteine (SFB) und System-funktionen (SFC)	SFBs und SFCs sind in die S7-CPU integriert und machen Ihnen einige wichtige Systemfunktionen zugänglich.	"Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)"
Funktionsbausteine (FB)	FBs sind Bausteine mit “Gedächtnis”, die Sie selbst programmieren.	"Funktionsbausteine (FB)"
Funktionen (FC)	FCs enthalten Programmroutinen für häufig verwendete Funktionen.	"Funktionen (FC)"
Instanz-Datenbausteine (Instanz-DB)	Instanz-DBs werden bei Aufruf eines FB/SFB dem Baustein zugeordnet. Beim Übersetzen werden sie automatisch generiert.	"Instanz-Datenbausteine"
Datenbausteine (DB)	DBs sind Datenbereiche zur Speicherung von Anwenderdaten. Zusätzlich zu den Daten, die jeweils einem Funktionsbaustein zugeordnet sind, können globale Daten definiert und von beliebigen Bausteinen genutzt werden.	"Globale Datenbausteine (DB)"

OBs, FBs, SFBs, FCs und SFCs enthalten Programmteile und werden deshalb auch als Codebausteine bezeichnet. Die erlaubte Anzahl der Bausteine pro Bausteintyp und die erlaubte Länge der Bausteine ist CPU-abhängig.

4.2.2 Organisationsbausteine und Programmstruktur

Organisationsbausteine (OBs) bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. Sie werden vom Betriebssystem aufgerufen und steuern die zyklische und alarmgesteuerte Programmbearbeitung, das Anlaufverhalten des Automatisierungssystems und die Behandlung von Fehlern. Sie können die Organisationsbausteine programmieren und so das Verhalten der CPU bestimmen.

Priorität von Organisationsbausteinen

Organisationsbausteine bestimmen die Reihenfolge (Startereignisse), in der die einzelnen Programmteile bearbeitet werden. Die Bearbeitung eines OB kann durch den Aufruf eines anderen OB unterbrochen werden. Welcher OB einen anderen OB unterbrechen darf, hängt von seiner Priorität ab. Höherpriore OBs unterbrechen niederpriore OBs. Die niedrigste Priorität hat der Hintergrund-OB.

Alarmarten und Prioritätsklassen

Die Startereignisse, die den Aufruf eines bestimmten OB veranlassen, werden auch Alarme genannt. Nachfolgende Tabelle zeigt die Alarmarten bei STEP 7 und die Priorität der zugeordneten Organisationsbausteinen. Nicht alle angegebenen Organisationsbausteine und deren Prioritätsklassen sind in allen S7-CPUs vorhanden (siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten").

Alarmart	Organisations-baustein	Prioritätsklasse (voreingestellt)	Näheres siehe
Freier Zyklus	OB 1	1	"Organisationsbaustein für zyklische Programmbearbeitung (OB 1)"
Uhrzeitalarme	OB 10 bis OB 17	2	"Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)"
Verzögerungsalarme	OB 20 OB 21 OB 22 OB 23	3 4 5 6	"Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)"
Weckalarme	OB 30 OB 31 OB 32 OB 33 OB 34 OB 35 OB 36 OB 37 OB 38	7 8 9 10 11 12 13 14 15	"Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)"
Prozeßalarme	OB 40 OB 41 OB 42 OB 43 OB 44 OB 45 OB 46 OB 47	16 17 18 19 20 21 22 23	"Organisationsbausteine für den Prozeßalarm(OB 40 bis OB 47)"
Multicomputingalarm	OB 60 Multicomputing	25	Multicomputing - Synchroner Betrieb mehrerer CPUs

Alarmart	**Organisations-baustein**	**Prioritätsklasse (voreingestellt)**	**Näheres siehe**
Redundanzfehler	OB 70 Peripherie-Redundanzfehler (nur in H-Systemen) OB 72 CPU-Redundanzfehler (nur in H-Systemen) Kommunikations-Redundanzfehler OB 73	25 28	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"
Asynchrone Fehler	OB 80 Zeitfehler OB 81 Stromversor-gungsfehler OB 82 Diagnosealarm OB 83 Ziehen/Stecken-Alarm OB 84 CPU-Hardwarefehler OB 85 Programmablauf-fehler OB 86 Baugruppenträger-ausfall OB 87 Kommunikations-fehler	26 (bzw. 28, wenn der asynchrone Fehler-OB im Anlaufprogramm vorkommt)	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"
Hintergrundzyklus	OB 90	29 [1)]	"Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)"
Anlauf	OB 100 Neustart OB 101 Wiederanlauf OB 102 Kaltstart	27 27 27	"Organisationsbausteine für den Anlauf (OB 100/OB 101/OB 102)"
Synchrone Fehler	OB 121 Programmierfehler OB 122 Zugriffsfehler	Prioritätsklasse des fehlerverursachen-den OB	"Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)"

1) Der Prioritätsklasse 29 entspricht die Priorität 0.29. Der Hintergrundzyklus hat also eine niedrigere Priorität als der Freie Zyklus.

Verändern der Priorität

Alarme sind mit STEP 7 parametrierbar. Mit der Parametrierung können Sie z. B. Alarm-OBs abwählen oder Prioritätsklassen in den Parameterblöcken: Uhrzeitalarme, Verzögerungsalarme, Weckalarme und Prozeßalarme ändern.

Bei S7-300-CPUs ist die Priorität der Organisationsbausteine fest zugeordnet.

Bei S7-400-CPUs (und der CPU 318) können Sie mit STEP 7 die Priorität folgender Organisationsbausteine ändern:

- OB 10 bis OB 47
- OB 70 bis OB 72 (nur H-CPUS) und OB 81 bis OB 87 im Betriebszustand RUN.

Zulässig sind:

- für OB 10 bis OB 47 die Prioritätsklassen 2 bis 23,
- für OB 70 bis OB 72 die Prioritätsklassen 2 bis 28 sowie
- für OB 81 bis OB 87 die Prioritätsklassen 24 bis 26.

Sie können mehreren OBs die gleiche Priorität zuteilen. OBs mit gleicher Priorität werden in der Reihenfolge des Auftretens ihrer Startereignisse bearbeitet.

Fehler-OBs, die bei synchronen Fehlern gestartet werden, werden in derselben Prioritätsklasse bearbeitet wie der Baustein, der beim Erkennen des Fehlers bearbeitet wurde.

Lokaldaten

Beim Erstellen von Codebausteinen (OBs, FCs, FBs) können Sie temporäre Lokaldaten vereinbaren. Der in der CPU zur Verfügung stehende Lokaldatenbereich wird unter den Prioritätsklassen aufgeteilt.

Bei S7-400-CPUs können Sie mit STEP 7 im Parameterblock ”Prioritätsklassen” die Anzahl der Lokaldaten pro Prioritätsklasse abändern.

Startinformation eines OB

Jeder Organisationsbaustein verfügt über eine Startinformation von 20 Byte Lokaldaten, die das Betriebssystem beim Starten eines OB übergibt. Die Startinformation gibt Auskunft über das Startereignis des OB, Datum und Uhrzeit des OB-Starts, aufgetretene Fehler und Diagnoseereignisse.

Ein Prozeßalarm-OB 40 enthält z. B. in der Startinformation die Adresse der alarmerzeugenden Baugruppe.

Abgewählte Alarm-OBs

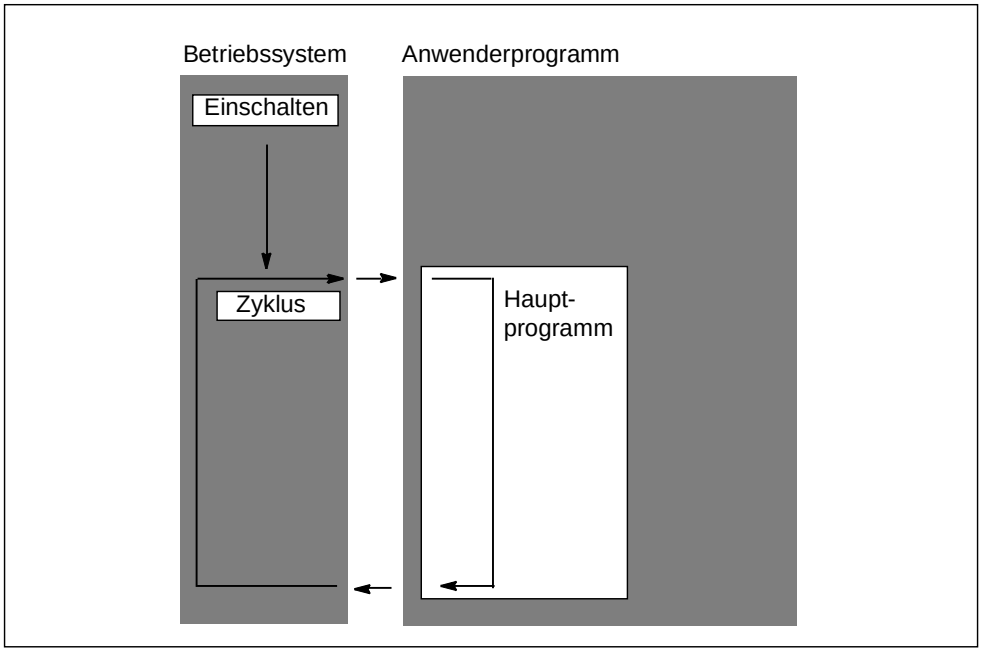
Wählen Sie die Prioritätsklassen 0 oder ordnen Sie einer Prioritätsklasse weniger als 20 Byte Lokaldaten zu, wird der entsprechende Alarm-OB abgewählt. Abgewählte Alarm-OBs können:

- im Betriebszustand "RUN" nicht kopiert bzw. in Ihr Anwenderprogramm eingebunden werden.
- im Betriebszustand "STOP" zwar kopiert bzw. in Ihr Anwenderprogramm eingebunden werden, führen jedoch beim Neustart der CPU zum Abbruch des Anlaufs und erzeugen einen Eintrag in den Diagnosepuffer.

Durch das Abwählen nicht benötigter Alarm-OBs wird der zur Verfügung stehende freie Lokaldatenbereich größer und kann für die Speicherung temporärer Daten in anderen Prioritätsklassen genutzt werden.

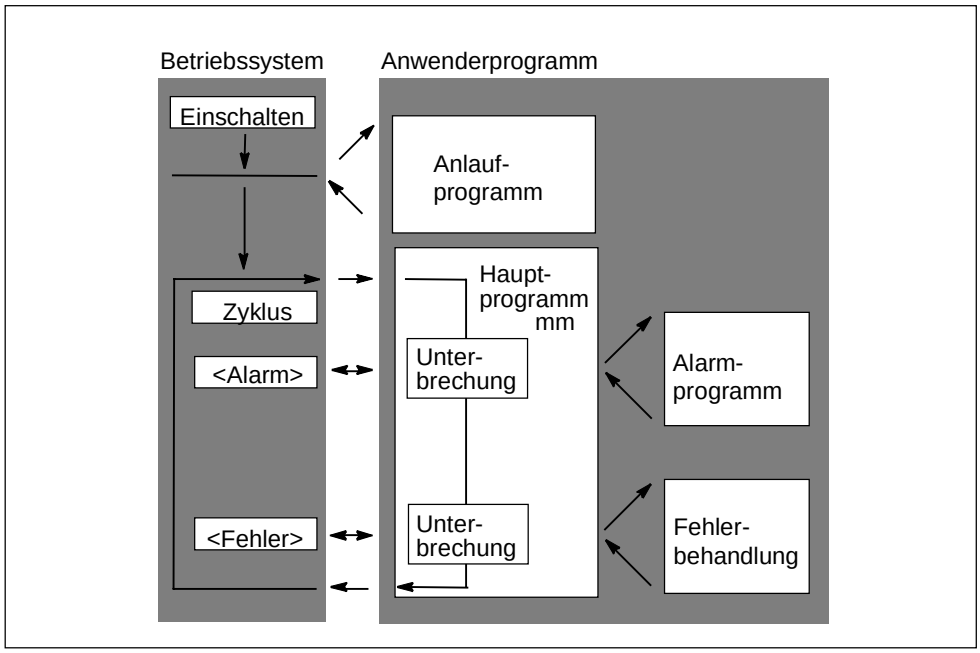
Zyklische Programmbearbeitung

Bei speicherprogrammierbaren Steuerungen herrscht typischerweise die zyklische Programmbearbeitung vor, d. h. das Betriebssystem läuft in einer Programmschleife (dem Zyklus) und ruft dabei in jeder Schleife einmal den Organisationsbaustein OB 1 im Hauptprogramm auf. Das Anwenderprogramm im OB 1 wird also zyklisch bearbeitet.



Ereignisgesteuerte Programmbearbeitung

Die zyklische Programmbearbeitung kann durch bestimmte Startereignisse (Alarme) unterbrochen werden. Tritt ein solches Startereignis ein, wird der gerade bearbeitete Baustein an einer Befehlsgrenze unterbrochen und ein anderer Organisationsbaustein abgearbeitet, der dem Startereignis zugeordnet ist. Danach wird die Bearbeitung des zyklischen Programms an der Unterbrechungsstelle wieder fortgesetzt.

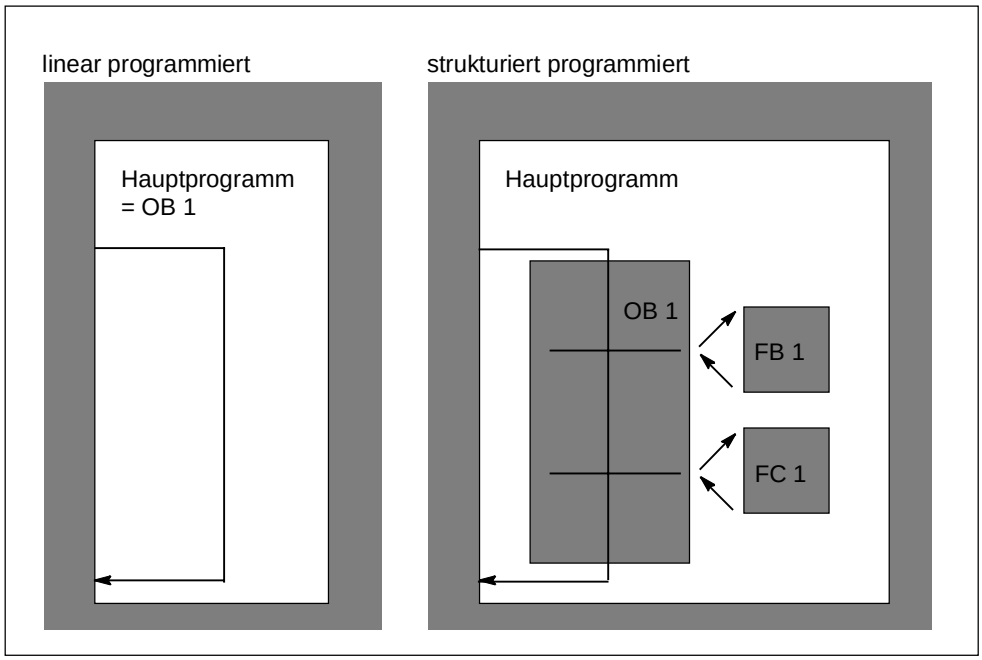


Damit besteht die Möglichkeit, Teile des Anwenderprogramms, die nicht zyklisch bearbeitet werden müssen, nur bei Bedarf zu bearbeiten. Das Anwenderprogramm kann in Teilprogramme zerlegt und auf verschiedene Organisationsbausteine aufgeteilt werden. Soll das Anwenderprogramm auf ein wichtiges Signal reagieren, das relativ selten vorkommt (z. B. Grenzwertgeber zur Füllstandsmessung eines Tanks meldet Füllstand erreicht), kann das Teilprogramm, das bearbeitet werden soll, wenn das Signal ausgegeben wird, in einem OB stehen, der ereignisgesteuert bearbeitet wird.

Linear oder strukturiert programmieren

Sie können Ihr gesamtes Anwenderprogramm in den OB 1 schreiben (lineare Programmierung). Dies ist nur empfehlenswert bei einfachen Programmen, die auf S7-300-CPUs mit kleinem Speicherumfang laufen.

Komplexe Automatisierungsaufgaben können besser bearbeitet werden, wenn sie in kleinere Teilaufgaben untergliedert werden, die den technologischen Funktionen des Automatisierungsprozesses entsprechen oder mehrfach verwendet werden sollen. Im Anwenderprogramm werden diese Teilaufgaben durch entsprechende Programmteile, die Bausteine, repräsentiert (strukturierte Programmierung).



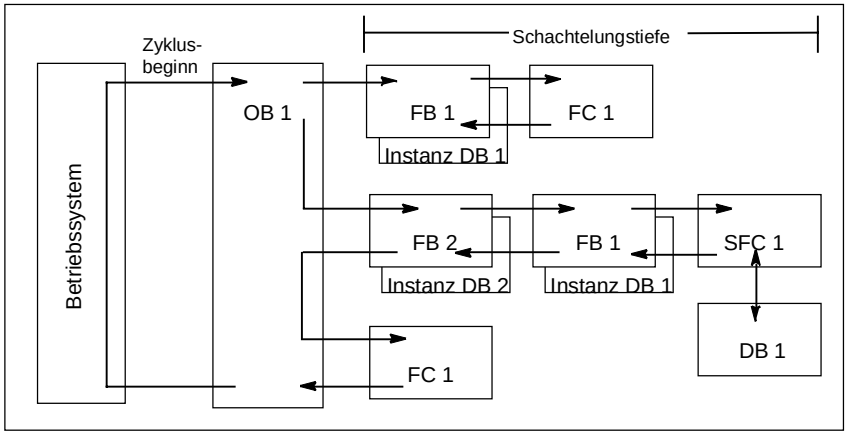
4.2.3 Aufrufhierarchie im Anwenderprogramm

Soll das Anwenderprogramm funktionieren, so müssen die Bausteine, aus denen es besteht, aufgerufen werden. Das geschieht durch spezielle STEP 7-Operationen, den Bausteinaufrufen, die nur innerhalb von Codebausteinen programmiert und gestartet werden können.

Reihenfolge und Schachtelungstiefe

Die Reihenfolge und Schachtelung der Bausteinaufrufe wird Aufrufhierarchie genannt. Die zulässige Schachtelungstiefe ist CPU-abhängig.

Nachfolgendes Bild zeigt anhand eines Beispiels die Reihenfolge und Schachtelung der Bausteinaufrufe innerhalb eines Bearbeitungszyklus.



Regeln für die Reihenfolge der Bausteinerstellung:

- Sie erstellen die Bausteine von oben nach unten, fangen also mit der obersten Bausteinreihe an.
- Jeder Baustein der aufgerufen wird, muß schon vorhanden sein, d.h. innerhalb einer Bausteinreihe ist die Erstellrichtung von rechts nach links.
- Als letztes erstellen Sie den OB 1.

Diese Regeln umgesetzt bedeuten für das Beispiel-Bild folgende Erstellreihenfolge:

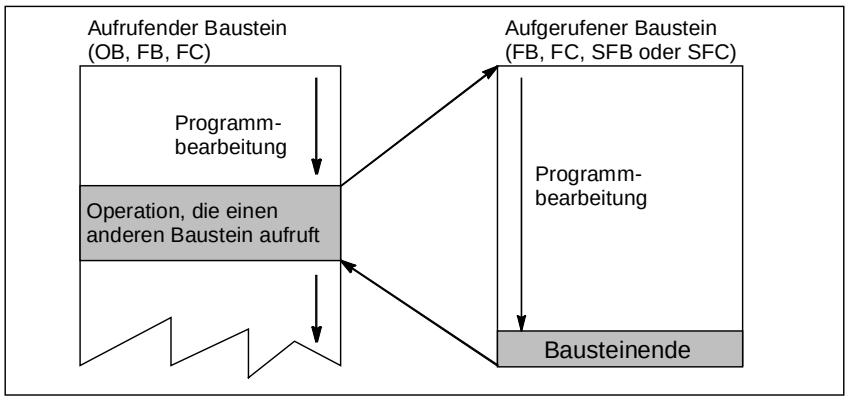
FC 1 > FB 1 + Instanz-DB 1 > DB 1 > SFC 1 > FB 2 + Instanz-DB 2 > OB 1

Hinweis

Bei zu großer Schachtelungstiefe kann der Lokaldaten-Stack überlaufen (siehe auch Lokaldaten-Stack).

Bausteinaufrufe

Nachfolgendes Bild zeigt den Ablauf eines Bausteinaufrufs innerhalb eines Anwenderprogramms: das Programm ruft den zweiten Baustein auf, dessen Operationen dann vollständig bearbeitet werden. Ist die Bearbeitung des aufgerufenen Bausteins beendet, wird die Bearbeitung des aufrufenden Bausteins mit der dem Bausteinaufruf folgenden Operation wiederaufgenommen.



Bevor Sie einen Baustein programmieren, müssen Sie erst festlegen, mit welchen Daten die Bearbeitung Ihres Programms erfolgen soll: Sie müssen die Variablen des Bausteins deklarieren.

Hinweis

OUT-Parameter müssen bei jedem Bausteinaufruf beschrieben werden.

Hinweis

Das Betriebssystem setzt die Instanzen des SFB 3 "TP" bei Kaltstart zurück. Falls Instanzen dieses SFB nach Neustart (Warmstart) initialisiert sein sollen, müssen Sie über den OB 100 die zu initialisierenden Instanzen des SFB mit PT = 0 ms aufrufen. Dies können Sie z.B. über eine Initialisierungsroutine in den Bausteinen erreichen, die Instanzen dieses SFB enthalten.

4.2.4 Bausteinarten und zyklische Programmbearbeitung

4.2.4.1 Organisationsbaustein für zyklische Programmbearbeitung (OB 1)

Die zyklische Programmbearbeitung ist die ”normale” Programmbearbeitung bei speicherprogrammierbaren Steuerungen. Das Betriebssystem ruft zyklisch den OB 1 auf und startet damit die zyklische Bearbeitung des Anwenderprogramms.

Ablauf der zyklischen Programmbearbeitung

Nachfolgende Tabelle zeigt die Phasen der zyklischen Programmbearbeitung:

Schritt	Ablauf bei bisherigen CPUs	Ablauf bei neuen CPUs (ab 10/98)
1	Das Betriebssystem startet die Zyklus-überwachungszeit.	Das Betriebssystem startet die Zyklus-überwachungszeit.
2	Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen aus und aktualisiert das Prozeßabbild der Eingänge.	Die CPU schreibt die Werte aus dem Prozeßabbild der Ausgänge in die Ausgabe-baugruppen.
3	Die CPU bearbeitet das Anwender-programm und führt die im Programm ange-gebenen Operationen aus.	Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen aus und aktualisiert das Prozeßabbild der Eingänge.
4	Die CPU schreibt die Werte aus dem Prozeßabbild der Ausgänge in die Ausgabe-baugruppen.	Die CPU bearbeitet das Anwender-programm und führt die im Programm ange-gebenen Operationen aus.
5	Am Ende eines Zyklus führt das Betriebs-system anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen, Empfangen und Senden von Globaldaten.	Am Ende eines Zyklus führt das Betriebs-system anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen, Empfangen und Senden von Globaldaten.
6	Anschließend kehrt die CPU zum Zyklusanfang zurück und startet erneut die Zykluszeitüberwachung.	Anschließend kehrt die CPU zum Zyklusanfang zurück und startet erneut die Zykluszeitüberwachung.

Prozeßabbilder

Damit der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozeßsignale zur Verfügung steht, greift die CPU beim Ansprechen der Operandenbereiche Eingänge (E) und Ausgänge (A) nicht direkt auf die Signalbaugruppen zu, sondern auf einen internen Speicherbereich der CPU, in dem sich ein Abbild der Ein-/Ausgänge befindet.

Programmieren zyklischer Programmbearbeitung

Sie programmieren die zyklische Programmbearbeitung, indem Sie Ihr Anwenderprogramm mit STEP 7 in den OB 1 und die darin aufgerufenen Bausteine schreiben.

Die zyklische Programmbearbeitung beginnt, sobald das Anlaufprogramm fehlerfrei beendet wurde.

Unterbrechungsmöglichkeiten

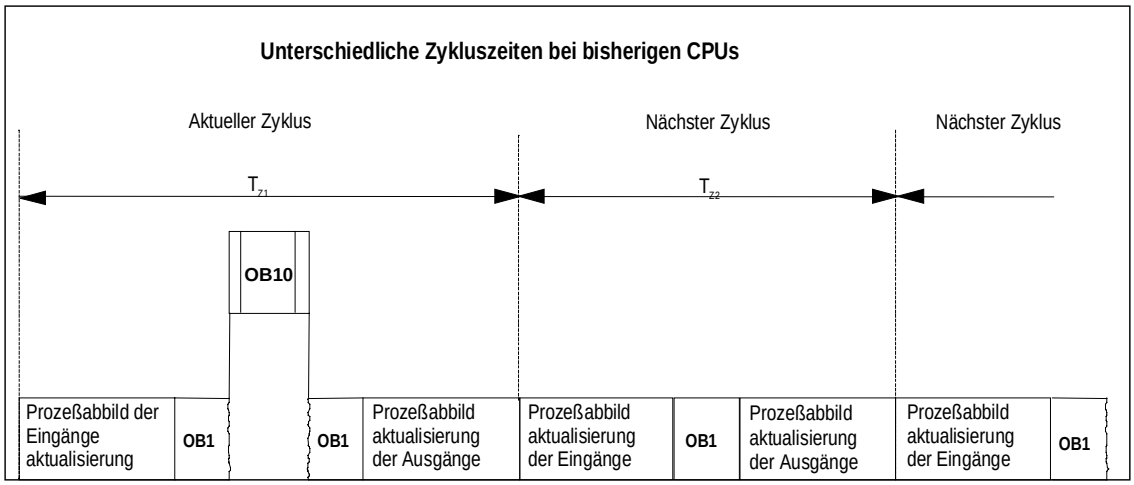
Die zyklische Programmbearbeitung kann unterbrochen werden durch:

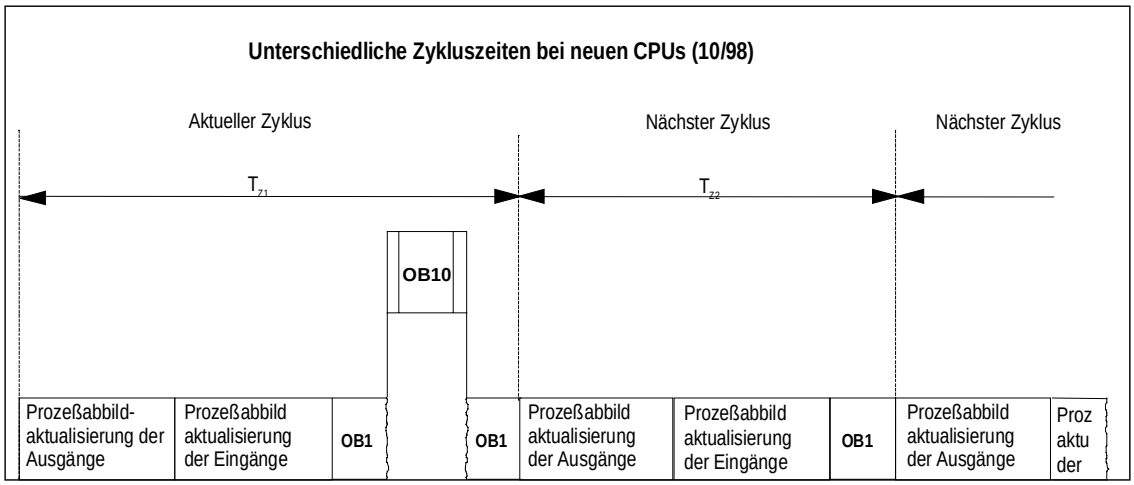
- einen Alarm
- einen STOP-Befehl (Betriebsartenschalter, Menübefehl vom PG aus, SFC 46 STP, SFB 20 STOP)
- Netzspannungsausfall
- das Auftreten eines Geräte- oder Programmfehlers

Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem für die Bearbeitung des zyklischen Programms sowie aller diesen Zyklus unterbrechenden Programmteile (z. B. Bearbeitung anderer Organisationsbausteine) und Systemtätigkeiten (z. B. Prozeßabbildaktualisierung) benötigt. Diese Zeit wird überwacht.

Die Zykluszeit (TZ) ist nicht für jeden Zyklus gleich lang. Nachfolgende Bilder zeigen unterschiedliche Zykluszeiten (TZ1 $\neq$ TZ2) für bisherige und neue CPUs.





Im aktuellen Zyklus wird OB 1 durch einen Uhrzeit-Alarm unterbrochen.

Maximalzykluszeit

Sie können mit STEP 7 die voreingestellte Maximalzykluszeit ändern. Ist diese Zeit abgelaufen, geht entweder die CPU in STOP, oder es wird der OB 80 aufgerufen, in dem Sie festlegen können, wie die CPU auf den Zeitfehler reagieren soll.

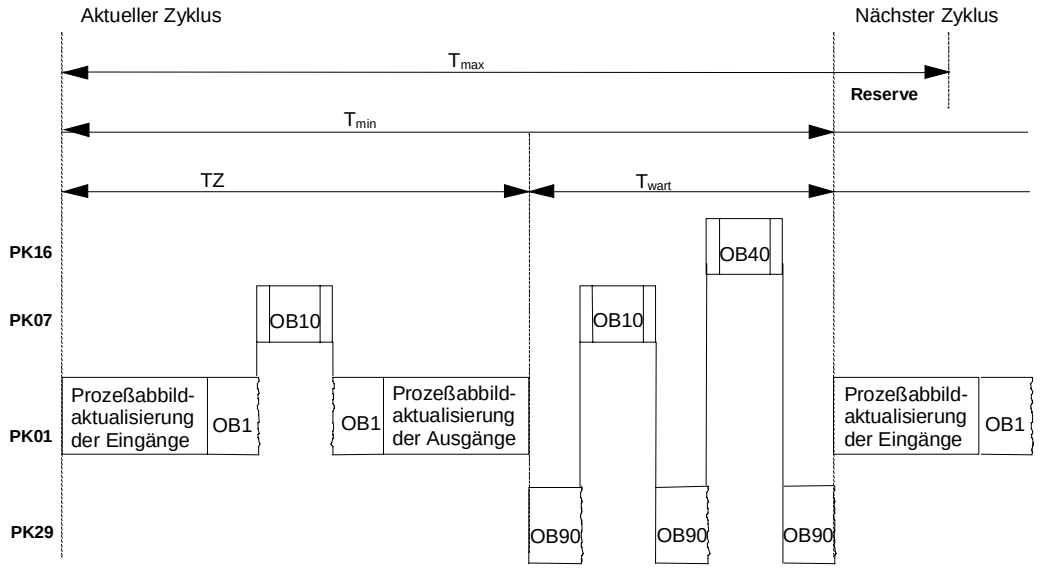
Mindestzykluszeit

Sie können mit STEP 7 bei S7-400-CPUs und bei der CPU 318 eine Mindestzykluszeit einstellen. Dies ist sinnvoll, wenn

- die Zeitabstände zwischen den Starts der Programmbearbeitung des OB 1 (Freier Zyklus) gleich lang sein sollen oder
- bei zu kurzer Zykluszeit die Aktualisierung der Prozeßabbilder unnötig oft erfolgen würde.

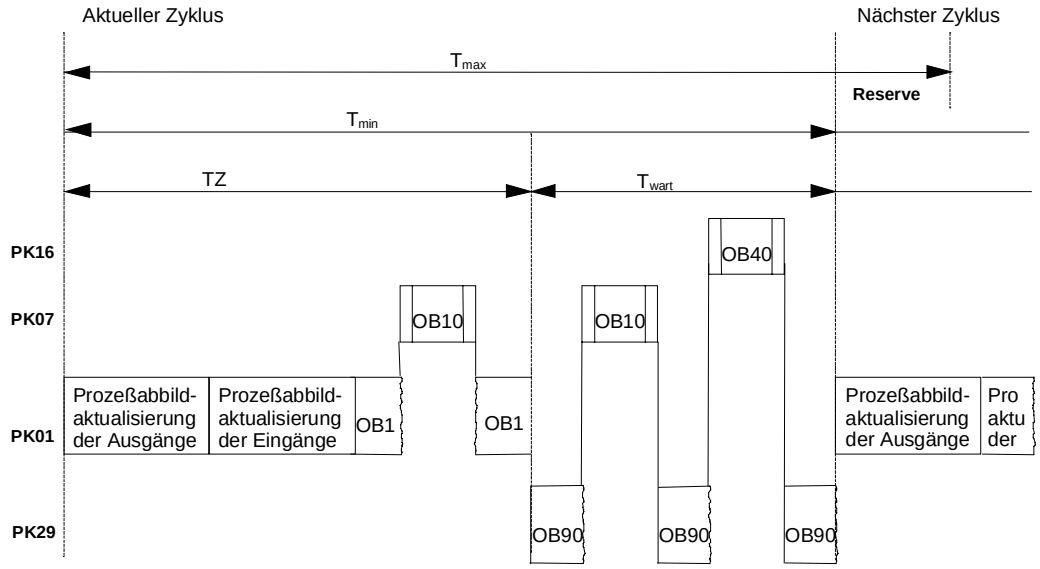
Nachfolgende Bilder zeigen die Funktion der Zyklusüberwachungszeit im Programmablauf bei den bisherigen und bei den neuen CPUs.

Zyklusüberwachungszeit bei bisherigen CPUs



Tmax = die einstellbare Maximalzykluszeit
Tmin = die einstellbare Mindestzykluszeit
Tz = die tatsächliche Zykluszeit
Twart = die Differenz zwischen T_{min} und tatsächlicher Zykluszeit, in dieser Zeit können auftretende Alarme sowie der Hintergrund-OB bearbeitet werden
PK = Prioritätsklasse

Zyklusüberwachungszeit bei neuen CPUs (10/98)



Tmax = die einstellbare Maximalzykluszeit
Tmin = die einstellbare Mindestzykluszeit
Tz = die tatsächliche Zykluszeit
Twart = die Differenz zwischen T_{min} und tatsächlicher Zykluszeit, in dieser Zeit können auftretende Alarme sowie der Hintergrund-OB bearbeitet werden
PK = Prioritätsklasse

Aktualisierung des Prozeßabbilds

Bei der zyklischen Programmbearbeitung der CPU wird automatisch das Prozeßabbild aktualisiert. Sie können bei S7-400-CPUs und bei der CPU 318 die Aktualisierung des Prozeßabbildes abwählen, wenn Sie

- statt dessen direkt auf die Peripherie zugreifen wollen oder
- ein oder mehrere Prozeßabbilder der Ein- und Ausgänge zu einem anderen Zeitpunkt mit Hilfe der Systemfunktionen SFC 26 UPDAT_PI und SFC 27 UPDAT_PO aktualisieren wollen.

Kommunikationslast

Mit dem CPU-Parameter "Zyklusbelastung durch Kommunikation" können Sie die Dauer von Kommunikationsprozessen, die immer auch die Zykluszeit verlängern, in einem gewissen Rahmen steuern. Kommunikationsprozesse können z. B. sein: Datenübertragung zu einer anderen CPU über MPI oder Laden von Bausteinen, das über PG angestoßen wurde.

Testfunktionen mit dem PG werden durch diesen Parameter kaum beeinflußt. Sie können aber die Zykluszeit erheblich verlängern. Die für Testfunktionen zur Verfügung gestellte Zeit kann im Prozeßbetrieb begrenzt werden (nur S7-300).

Wirkungsweise des Parameters

Das Betriebssystem der CPU stellt laufend der Kommunikation den projektierten Prozentsatz der gesamten CPU-Verarbeitungsleistung zur Verfügung (Zeitscheiben-Technik). Wird diese Verarbeitungsleistung für die Kommunikation nicht benötigt, steht sie der übrigen Verarbeitung zur Verfügung.

Auswirkung auf die tatsächliche Zykluszeit

Ohne zusätzliche asynchrone Ereignisse verlängert sich die OB 1-Zykluszeit um einen Faktor, der sich nach folgender Formel berechnen läßt:

$$\frac{100}{100 - \text{"Zyklusbelastung durch Kommunikation (\%)"}}$$

Beispiel 1 (keine zusätzlichen asynchrone Ereignisse):

Bei Einstellung der Zyklusbelastung durch Kommunikation auf 50 % kann sich eine Verdopplung der OB 1-Zykluszeit ergeben.

Gleichzeitig wird die OB 1-Zykluszeit auch noch durch asynchrone Ereignisse (z. B. Prozeßalarme oder Weckalarme) beeinflußt. Durch die Verlängerung der Zykluszeit durch den Kommunikationsanteil treten statistisch gesehen auch mehr asynchrone Ereignisse innerhalb eines OB 1-Zyklus auf. Dies verlängert der OB 1-Zyklus zusätzlich. Diese Verlängerung ist abhängig davon, wieviele Ereignisse pro OB 1-Zyklus auftreten und der Dauer der Ereignisbearbeitung.

Beispiel 2 (zusätzlichen asynchrone Ereignisse berücksichtigt):

Bei einer reinen OB 1-Ablaufzeit von 500 ms kann sich durch eine Kommunikationslast von 50 % eine tatsächliche Zykluszeit von bis zu 1000 ms ergeben (unter der Voraussetzung, daß die CPU immer genügend Kommunikationsaufträge zum Bearbeiten hat). Wenn nun parallel dazu alle 100 ms ein Weckalarm mit 20 ms Bearbeitungszeit abläuft, dann würde sich dieser ohne Kommunikationslast mit insgesamt 5*20 ms = 100 ms auf den Zyklus verlängernd auswirken, d. h. die tatsächliche Zykluszeit wäre 600 ms. Da ein Weckalarm auch die Kommunikation unterbricht, wirkt er sich bei 50% Kommunikationslast mit 10 * 20 ms auf die Zykluszeit aus, d.h. in diesem Fall beträgt die tatsächliche Zykluszeit nicht 1000 ms sondern 1200 ms.

Hinweise

- Überprüfen Sie die Auswirkungen einer Wertänderung des Parameters "Zyklusbelastung durch Kommunikation" im Anlagenbetrieb.
- Die Kommunikationslast muß beim Einstellen der minimalen Zykluszeit berücksichtigt werden, da es sonst zu Zeitfehlern kommt.

Empfehlungen

- Übernehmen Sie nach Möglichkeit den voreingestellten Wert
- Vergrößern Sie den Wert nur dann, wenn die CPU hauptsächlich zu Kommunikationszwecken eingesetzt wird und das Anwenderprogramm zeitunkritisch ist!
- In allen anderen Fällen den Wert nur verringern!
- Stellen Sie Prozeßbetrieb ein (nur bei S7-300) und begrenzen Sie die für Testfunktionen dort benötigte Zeit!

4.2.4.2 Funktionen (FC)

Funktionen gehören zu den Bausteinen, die Sie selbst programmieren. Eine Funktion ist ein Codebaustein ”ohne Gedächtnis”. Temporäre Variablen der FC werden im Lokaldaten-Stack gespeichert. Diese Daten gehen nach der Bearbeitung der FC verloren. Funktionen können zum Speichern von Daten globale Datenbausteine nutzen.

Weil eine FC keinen zugeordneten Speicher hat, müssen Sie immer Aktualparameter für eine FC angeben. Sie können den Lokaldaten einer FC keine Anfangswerte zuordnen.

Anwendungsbereich

Eine FC enthält ein Programm, das immer dann ausgeführt wird, wenn die FC von einem anderen Codebaustein aufgerufen wird. Funktionen können eingesetzt werden, um

- einen Funktionswert an den aufrufenden Baustein zurückzugeben (Beispiel: mathematische Funktionen).
- eine technologische Funktion auszuführen (Beispiel: Einzelsteuerung mit Binärverknüpfung).

Zuordnen von Aktualparametern zu Formalparametern

Ein Formalparameter ist ein Platzhalter für den “tatsächlichen” Parameter, den Aktualparameter. Aktualparameter ersetzen beim Aufruf einer FC die Formalparameter. Sie müssen den Formalparametern einer FC immer Aktualparameter zuordnen (z.B. dem Formalparameter “Start” einen Aktualparameter “E3.6”). Die Eingangs-, Ausgangs- und Durchgangsparameter, die von der FC verwendet werden, werden als Pointer auf die Aktualparameter des Codebausteins gespeichert, der die FC aufgerufen hat.

4.2.4.3 Funktionsbausteine (FB)

Funktionsbausteine gehören zu den Bausteinen, die Sie selbst programmieren. Ein Funktionsbaustein ist ein Baustein ”mit Gedächtnis”. Er verfügt über einen zugeordneten Datenbaustein als Speicher (Instanz-Datenbaustein). Die Parameter, die an den FB übergeben werden, sowie die statischen Variablen werden im Instanz-DB gespeichert. Die temporären Variablen werden im Lokaldaten-Stack gespeichert.

Daten, die im Instanz-DB gespeichert werden, gehen nicht verloren, wenn die Bearbeitung des FB beendet ist. Daten, die im Lokaldaten-Stack gespeichert werden, gehen nach der Bearbeitung des FB verloren.

Hinweis

Um Fehler beim Arbeiten mit FBs zu vermeiden, lesen Sie Zulässige Datentypen beim Übergeben von Parametern im Anhang.

Anwendungsbereich

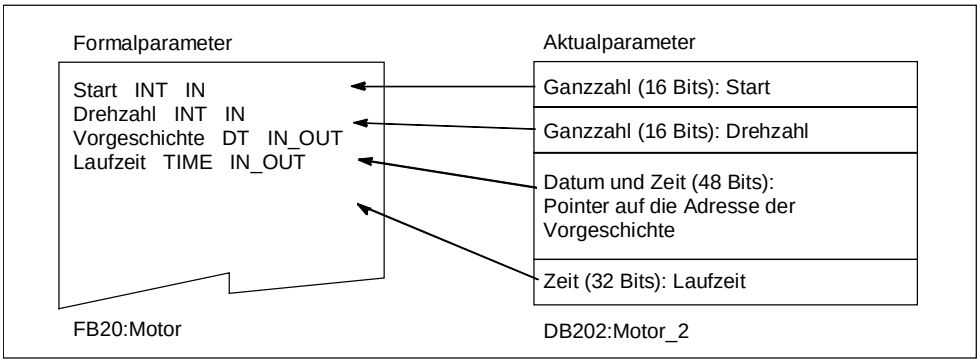
Ein FB enthält ein Programm, das immer dann ausgeführt wird, wenn der FB von einem anderen Codebaustein aufgerufen wird. Funktionsbausteine erleichtern das Programmieren häufig wiederkehrender, komplexer Funktionen.

FBs und Instanz-DBs

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet.

Durch das Aufrufen mehrerer Instanzen eines FB können Sie mit einem FB mehrere Geräte steuern. Ein FB für einen Motortyp beispielsweise kann verschiedene Motoren steuern, indem er verschiedene Instanz-Daten für die verschiedenen Motoren verwendet. Die Daten für jeden einzelnen Motor (wie Drehzahl, Ramping, akkumulierte Betriebszeit usw.) können in einem oder mehreren Instanz-DBs gespeichert werden.

Nachfolgendes Bild zeigt die Formalparameter eines FB, der die Aktualparameter verwendet, die im Instanz-DB gespeichert sind.



Variablen vom Datentyp FB

Ist Ihr Anwenderprogramm so strukturiert, daß in einem FB weitere, bereits existierende Funktionsbausteine aufgerufen werden, können Sie in die Variablendeklarationstabelle des aufrufenden FB die aufzurufenden FBs als statische Variablen mit dem Datentyp FB aufnehmen. Damit erreichen Sie eine Verschachtelung von Variablen und die Konzentrierung der Instanzdaten in einem Instanz-Datenbaustein (Multiinstanz).

Zuordnen von Aktualparametern zu Formalparametern

Im allgemeinen ist es in STEP 7 nicht erforderlich, daß Sie den Formalparametern eines FB Aktualparameter zuordnen. Es gibt jedoch Ausnahmen: Aktualparameter müssen zugeordnet werden:

- bei einem Durchgangsparameter eines zusammengesetzten Datentyps (z. B. STRING, ARRAY oder DATE_AND_TIME)
- bei allen Parametertypen (z. B. TIMER, COUNTER oder POINTER)

STEP 7 ordnet den Formalparametern eines FB die Aktualparameter folgendermaßen zu:

- *Wenn Sie in der Aufrufanweisung Aktualparameter angeben:* Die Operationen des FB verwenden die bereitgestellten Aktualparameter.
- *Wenn Sie in der Aufrufanweisung keine Aktualparameter angeben:* Die Operationen des FB verwenden die Werte, die in dem Instanz-DB gespeichert sind.

Nachfolgende Tabelle zeigt, welchen Variablen des FB Aktualparameter zugeordnet werden müssen.

Variablen		**Datentyp**	
	Elementarer Datentyp	**Zusammengesetzter Datentyp**	**Parametertyp**
Eingang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Ausgang	Parameter nicht erforderlich	Parameter nicht erforderlich	Aktualparameter erforderlich
Durchgang	Parameter nicht erforderlich	Aktualparameter erforderlich	–

Zuordnen von Anfangswerten zu Formalparametern

Sie können den Formalparametern im Deklarationsteil des FB Anfangswerte zuordnen. Diese Werte werden in den Instanz-DB übernommen, der einem FB zugeordnet wird.

Ordnen Sie den Formalparametern in der Aufrufanweisung keine Aktualparameter zu, dann verwendet STEP 7 die Werte, die im Instanz-DB gespeichert sind. Diese Daten können Anfangswerte sein, die in der Variablendeklarationstabelle eines FB eingegeben wurden.

Nachfolgende Tabelle zeigt, welche Variablen einem Anfangswert zugeordnet werden können. Weil die temporären Daten nach der Bearbeitung des Bausteins nicht gespeichert werden, können Sie ihnen keine Werte zuordnen.

	Datentyp		
Variablen	**Elementarer Datentyp**	**Zusammengesetzter Datentyp**	**Parametertyp**
Eingang	Anfangswert zulässig	Anfangswert zulässig	–
Ausgang	Anfangswert zulässig	Anfangswert zulässig	–
Durchgang	Anfangswert zulässig	–	–
Statisch	Anfangswert zulässig	Anfangswert zulässig	–
Temporär	–	–	–

4.2.4.4 Instanz-Datenbausteine

Jedem Aufruf eines Funktionsbausteins, der Parameter übergibt, ist ein Instanz-Datenbaustein zugeordnet. Im Instanz-DB sind die Aktualparameter und die statischen Daten des FB abgelegt. Die im FB deklarierten Variablen bestimmen die Struktur des Instanz-Datenbausteins. Mit Instanz wird der Aufruf eines Funktionsbausteins bezeichnet. Wird z. B. ein Funktionsbaustein im S7-Anwenderprogramm fünfmal aufgerufen, so existieren fünf Instanzen dieses Bausteins.

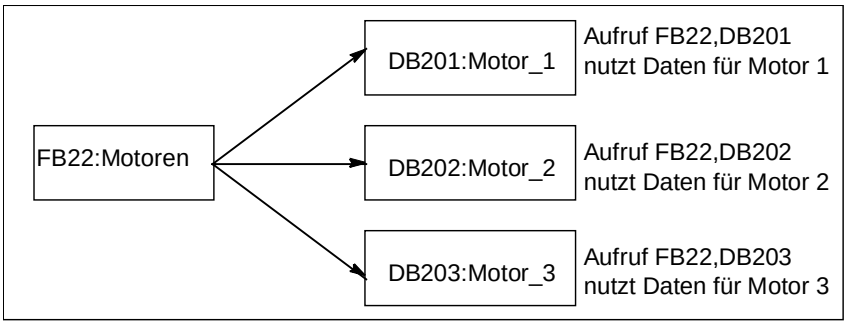
Erzeugen eines Instanz-DB

Bevor Sie einen Instanz-Datenbaustein erzeugen, muß der zuzuordnende FB bereits existieren. Die Nummer dieses FB geben Sie beim Anlegen des Instanz-Datenbausteins an.

Ein Instanz-DB für jede Instanz

Wenn Sie mehrere Instanz-Datenbausteine einem Funktionsbaustein (FB) zuordnen, der einen Motor steuert, so können Sie diesen FB zum Steuern verschiedener Motoren verwenden.

Die unterschiedlichen Daten für die einzelnen Motoren (wie z. B. Drehzahl, Hochlaufzeit, Gesamtbetriebszeit) werden in den verschiedenen Datenbausteinen gespeichert. Je nachdem welcher DB dem FB beim Aufruf zugeordnet wird, kann ein anderer Motor gesteuert werden. Auf diese Weise wird nur ein Funktionsbaustein für mehrere Motoren benötigt (siehe nachfolgendes Bild).

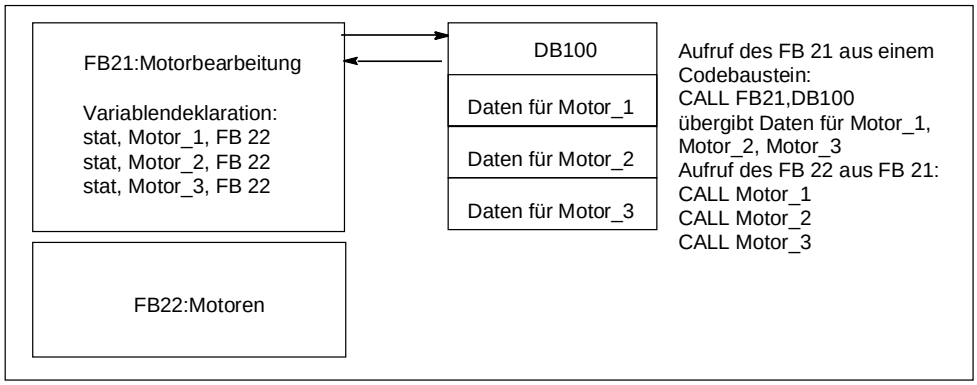


Ein Instanz-DB für mehrere Instanzen eines FB (Multiinstanzen)

Sie können einem FB die Instanzdaten für verschiedene Motoren gemeinsam in einem Instanz-DB übergeben. Dazu müssen Sie den Aufruf der Motorensteuerungen in einem weiteren FB vornehmen und im Deklarationsteil des rufenden FB statische Variablen mit dem Datentyp eines FB für die einzelnen Instanzen deklarieren.

Wenn Sie einen Instanz-DB für mehrere Instanzen eines FB verwenden, sparen Sie Speicherplatz und optimieren den Einsatz von Datenbausteinen.

Im nachfolgenden Bild ist beispielsweise der rufende FB der FB 21 ”Motorbearbeitung”, die Variablen sind vom Datentyp FB 22 und die Instanzen werden mit Motor_1, Motor_2 und Motor_3 bezeichnet.



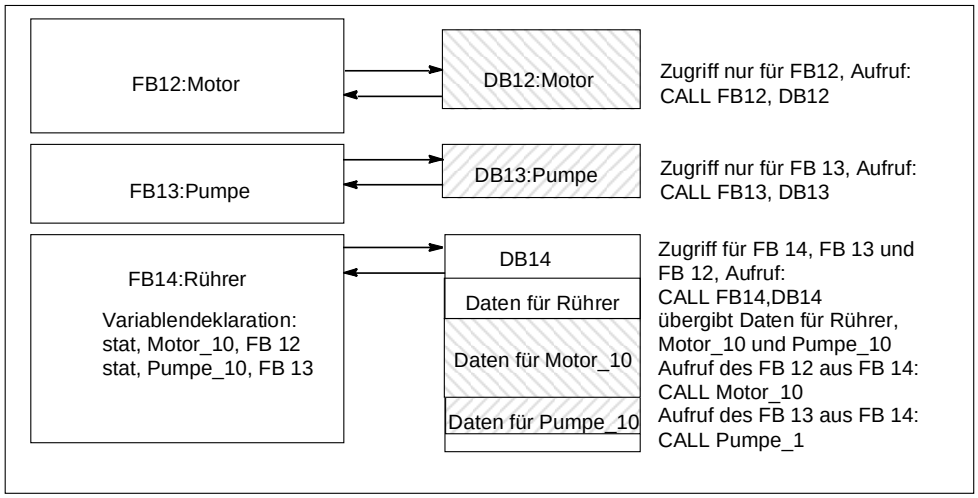
Der FB 22 braucht in diesem Beispiel keinen eigenen Instanz-Datenbaustein, weil seine Instanzdaten im Instanz-Datenbaustein des rufenden FB gespeichert sind.

Ein Instanz-DB für mehrere Instanzen unterschiedlicher FBs (Multiinstanzen)

Sie können in einem Funktionsbaustein Instanzen anderer, bereits erstellter FBs aufrufen. Die dafür benötigten Instanz-Daten können sie dem Instanz-Datenbaustein des aufrufenden FBs zuordnen, d. h. Sie brauchen für die aufgerufenen FBs in diesem Fall keine zusätzlichen Datenbausteine.

Für diese Multiinstanzen in einem Instanz-DB müssen Sie im Deklarationsteil des rufenden FBs statische Variablen mit dem Datentyp des aufgerufenen FBs für die einzelnen Instanzen deklarieren. Der Aufruf innerhalb des FBs erfolgt dann ohne Angabe eines Instanz-DBs nur über den Namen der Variable.

In dem Beispiel im Bild werden die zugeordneten Instanzdaten gemeinsam in einem Instanz-DB gespeichert.



4.2.4.5 Globale Datenbausteine (DB)

Datenbausteine enthalten im Gegensatz zu Codebausteinen keine STEP 7-Anweisungen. Sie dienen der Aufnahme von Anwenderdaten, d. h. in den Datenbausteinen stehen variable Daten, mit denen das Anwenderprogramm arbeitet. Globale Datenbausteine dienen der Aufnahme von Anwenderdaten, die von allen anderen Bausteinen aus verwendet werden können.

Die Größe von DBs kann variieren. Die maximal zulässige Größe entnehmen Sie den CPU-Beschreibungen /70/ und /101/.

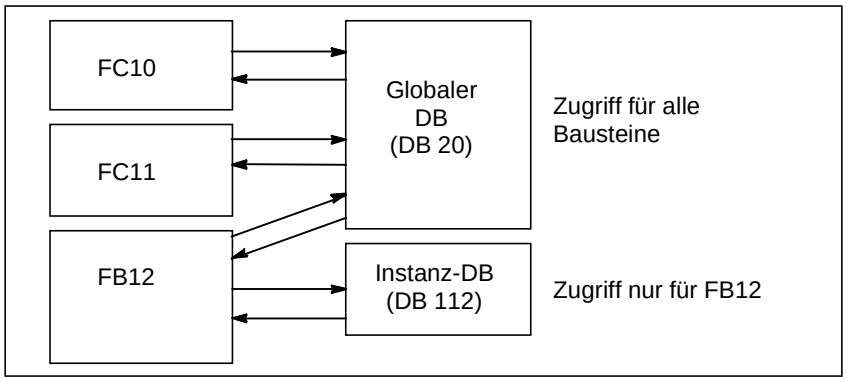
Die Struktur globaler Datenbausteine können Sie beliebig festlegen.

Globale Datenbausteine im Anwenderprogramm

Wird ein Codebaustein (FC, FB oder OB) aufgerufen, so kann er temporär Speicherplatz im Lokaldatenbereich (L-Stack) belegen. Zusätzlich zu diesem Lokaldatenbereich kann ein Codebaustein einen Speicherbereich in Form eines DB öffnen. Im Gegensatz zu den Daten im Lokaldatenbereich werden die in einem DB enthaltenen Daten nicht gelöscht, wenn der DB geschlossen wird, bzw. die Bearbeitung des zugehörigen Codebausteins beendet ist.

Jeder FB, FC oder OB kann die Daten aus einem globalen DB lesen oder selbst Daten in einen globalen DB schreiben. Diese Daten bleiben im DB auch dann erhalten, wenn der DB verlassen wird.

Ein globaler DB und ein Instanz-DB können gleichzeitig geöffnet sein. Nachfolgendes Bild zeigt die verschiedenen Zugriffe auf Datenbausteine.



4.2.4.6 Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC)

Vorgefertigte Bausteine

Sie müssen nicht jede Funktion selbst programmieren. S7-CPUs bieten Ihnen vorgefertigte Bausteine, die Sie aus dem Anwenderprogramm heraus aufrufen können.

Nähere Informationen finden Sie in der Referenzhilfe zu Systembausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute).

Systemfunktionsbausteine

Ein Systemfunktionsbaustein SFB ist ein Funktionsbaustein, der in die S7-CPU integriert ist. Weil SFBs Teil des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FBs sind SFBs Bausteine ”mit Gedächtnis”. Sie müssen auch für SFBs Instanz-Datenbausteine erstellen und als Teil des Programms in die CPU laden.

S7-CPUs bieten SFBs

- zur Kommunikation über projektierte Verbindungen
- für integrierte Sonderfunktionen (z. B. SFB 29 ”HS_COUNT” auf der CPU 312 IFM und der CPU 314 IFM)

Systemfunktionen

Eine Systemfunktion ist eine vorprogrammierte Funktion, die in die S7-CPU integriert ist. Sie können die SFC aus Ihrem Programm aufrufen. Weil SFCs Teile des Betriebssystems sind, werden sie nicht als Teil des Programms geladen. Wie FCs sind SFCs Bausteine ”ohne Gedächtnis”.

S7-CPUs bieten SFCs für:

- Kopier- und Bausteinfunktionen
- die Programmkontrolle
- die Hantierung der Uhr und des Betriebsstundenzählers
- die Übertragung von Datensätzen
- Übertragung von Ereignissen im Betriebsmodus Multicomputing, von einer CPU auf alle gesteckten CPUs
- die Hantierung von Uhrzeit- und Verzögerungsalarmen
- die Hantierung von Synchronfehlerereignissen, Alarm- und Asynchronfehlerereignissen
- die Auskunft über statische und dynamische Systemdaten, z.B. Diagnose

- die Prozeßabbildaktualisierung und die Bitfeldbearbeitung
- die Adressierung von Baugruppen
- die dezentrale Peripherie
- die Globaldaten-Kommunikation
- die Kommunikation über nichtprojektierte Verbindungen
- Erzeugung bausteinbezogener Meldungen

Weitere Informationen

Nähere Informationen zu SFBs und SFCs entnehmen Sie dem Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen". Dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" oder dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten" entnehmen Sie, welche SFBs und SFCs verfügbar sind.

4.2.5 Organisationsbausteine für alarmgesteuerte Programmbearbeitung

4.2.5.1 Organisationsbausteine für alarmgesteuerte Programmbearbeitung

Die S7-CPUs geben Ihnen durch die Bereitstellung von Alarm-OBs die Möglichkeit

- Programmteile zeitgesteuert zu bearbeiten
- auf externe Signale des Prozesses optimal zu reagieren.

Das zyklische Anwenderprogramm muß nicht ständig abfragen, ob Alarmereignisse eingetreten sind, sondern das Betriebssystem sorgt im Falle eines Alarms dafür, daß der Teil des Anwenderprogramms bearbeitet wird, der im Alarm-OB steht und festlegt, wie das Automatisierungssystems bei diesem Alarm reagieren soll.

Alarmarten und Anwendungen

Nachfolgende Tabelle zeigt, wie die Alarmarten eingesetzt werden können.

Alarmart	Alarm-OBs	Anwendungsbeispiele
Uhrzeitalarm	OB 10 bis OB 17	Errechnen der Durchflußmenge eines Mischprozesses bei Schichtende
Verzögerungsalarm	OB 20 bis OB 23	Steuern eines Lüfters, der nach dem Abschalten eines Motors noch 20 s laufen soll, bevor er abgeschaltet wird.
Weckalarm	OB 30 bis OB 38	Abtasten eines Signalpegels für eine regelungstechnische Anlage
Prozeßalarm	OB 40 bis OB 47	Melden, daß der maximale Füllstand eines Behälters erreicht ist.

4.2.5.2 Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17)

Die S7-CPUs stellen Uhrzeitalarm-OBs zur Verfügung, die zu einem angegebenen Datum oder in bestimmten Zeitabständen bearbeitet werden können.

Uhrzeitalarme können ausgelöst werden:

- einmalig zu einem bestimmten Zeitpunkt (Absolutzeitangabe mit Datum)
- periodisch mit Angabe des Startzeitpunktes und der Wiederholrate (z. B. minütlich, stündlich, täglich).

Regeln für Uhrzeitalarme

Uhrzeitalarme können nur bearbeitet werden, wenn ein Uhrzeitalarm parametriert wurde und ein entsprechender Organisationsbaustein im Anwenderprogramm enthalten ist. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Periodische Uhrzeitalarme müssen einem realen Datum entsprechen. Die monatliche Wiederholung eines OB 10 mit Startzeit 31.1. ist nicht möglich. In diesem Fall würde der OB nur in den Monaten gestartet, die 31 Tage haben.

Ein Uhrzeit-Alarm, der während des Anlaufs (Neustart oder Wiederanlauf) aktiviert wird, wird erst nach der Beendigung des Anlaufs bearbeitet.

Uhrzeitalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Nach Neustart müssen gestellte Uhrzeitalarme neu aktiviert werden (z. B. mit Hilfe der SFC 30 ACT_TINT im Anlaufprogramm).

Starten des Uhrzeitalarms

Damit die CPU einen Uhrzeitalarm starten kann, müssen Sie den Uhrzeitalarm erst stellen und dann aktivieren. Es gibt drei Startmöglichkeiten:

- automatischer Start des Uhrzeitalarms durch Parametrieren mit STEP 7 (Parameterblock ”Uhrzeitalarme”)
- stellen und aktivieren des Uhrzeitalarms durch SFC 28 SET_TINT und SFC 30 ACT_TINT aus dem Anwenderprogramm heraus
- stellen des Uhrzeitalarms durch Parametrieren mit STEP 7 und aktivieren des Uhrzeitalarms durch die SFC 30 ACT_TINT aus dem Anwenderprogramm heraus.

Abfragen des Uhrzeitalarms

Um abzufragen, ob und zu welchem Zeitpunkt Uhrzeitalarme gestellt sind, können Sie

- die SFC 31 QRY_TINT aufrufen oder
- die Teilliste "Alarmstatus" der Systemzustandsliste anfordern.

Deaktivieren des Uhrzeitalarms

Sie können Uhrzeitalarme, die noch nicht bearbeitet wurden, mit der SFC 29 CAN_TINT deaktivieren. Deaktivierte Uhrzeitalarme lassen sich mit der SFC 28 SET_TINT erneut stellen und mit der SFC 30 ACT_TINT aktivieren.

Priorität der Uhrzeitalarm-OBs

Alle acht Uhrzeitalarm-OBs haben in der Voreinstellung die gleiche Prioritätsklasse (2) und werden dementsprechend in der Reihenfolge ihres Startereignisses bearbeitet. Die Prioritätsklasse kann durch Parametrierung geändert werden.

Ändern der eingestellten Uhrzeit

Es gibt die folgenden Möglichkeiten, die eingestellte Uhrzeit zu ändern:

- ein Uhrzeitmaster synchronisiert die Uhrzeit für Master und Slaves
- im Anwenderprogramm wird durch SFC 0 SET_CLK die Uhrzeit neu gestellt.

Verhalten bei Uhrzeitänderung

Nachfolgende Tabelle zeigt, wie sich Uhrzeitalarme nach dem Ändern der Uhrzeit verhalten.

Wenn...	**dann...**
durch das Vorstellen der Uhr ein oder mehrere Uhrzeitalarme übersprungen werden,	wird der OB 80 gestartet und in die Start-information des OB 80 wird eingetragen, welche Uhrzeitalarme übersprungen wurden.
Sie im OB 80 die übersprungenen Uhrzeit-alarme deaktiviert haben,	werden die übersprungenen Uhrzeitalarme nicht nachgeholt.
Sie im OB 80 die übersprungenen Uhrzeit-alarme nicht deaktiviert haben,	wird der erste übersprungene Uhrzeitalarm nachgeholt, die anderen übersprungenen Uhrzeitalarme werden ignoriert.
durch das Rückstellen der Uhr bereits bearbeitete Uhrzeitalarme erneut anstehen,	wird die Bearbeitung dieser Uhrzeitalarme nicht wiederholt.

4.2.5.3 Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)

Die S7-CPUs stellen Verzögerungsalarm-OBs zur Verfügung, mit deren Hilfe Sie die zeitverzögerte Bearbeitung von Teilen Ihres Anwenderprogramms programmieren können.

Regeln für Verzögerungsalarme

Verzögerungsalarme können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (OB 80, siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Verzögerungsalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Verzögerungsalarme werden ausgelöst, wenn die in der SFC 32 SRT_DINT angegebene Verzögerungszeit abgelaufen ist.

Starten des Verzögerungsalarms

Um einen Verzögerungsalarm zu starten, müssen Sie in der SFC 32 die Verzögerungszeit festlegen, nach deren Ablauf der entsprechende Verzögerungsalarm-OB aufgerufen werden soll. Die maximal zulässige Länge der Verzögerungszeit entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten".

Priorität der Verzögerungsalarm-OBs

In der Voreinstellung haben die Verzögerungsalarm-OBs die Prioritätsklassen 3 bis 6. Die Prioritätsklassen können durch Parametrierung geändert werden.

4.2.5.4 Organisationsbausteine für den Weckalarm (OB 30 bis OB 38)

Die S7-CPUs stellen Weckalarm-OBs zur Verfügung, die die zyklische Programmbearbeitung in bestimmten Abständen unterbrechen.

Weckalarme werden in bestimmten Zeitintervallen ausgelöst. Startzeitpunkt des Zeittaktes ist der Betriebszustandswechsel von STOP in RUN.

Regeln für Weckalarme

Achten Sie beim Vorgeben der Zeittakte darauf, daß zwischen den Startereignissen der einzelnen Weckalarme genügend Zeit für die Bearbeitung der Weckalarme bleibt.

Weckalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Starten des Weckalarms

Um einen Weckalarm zu starten, müssen Sie mit STEP 7 im Parameterblock “Weckalarme” einen Zeittakt vorgeben. Der Zeittakt ist immer ein ganzzahliges Vielfaches des Grundtaktes von 1 ms.

Zeittakt = n x Grundtakt 1 ms

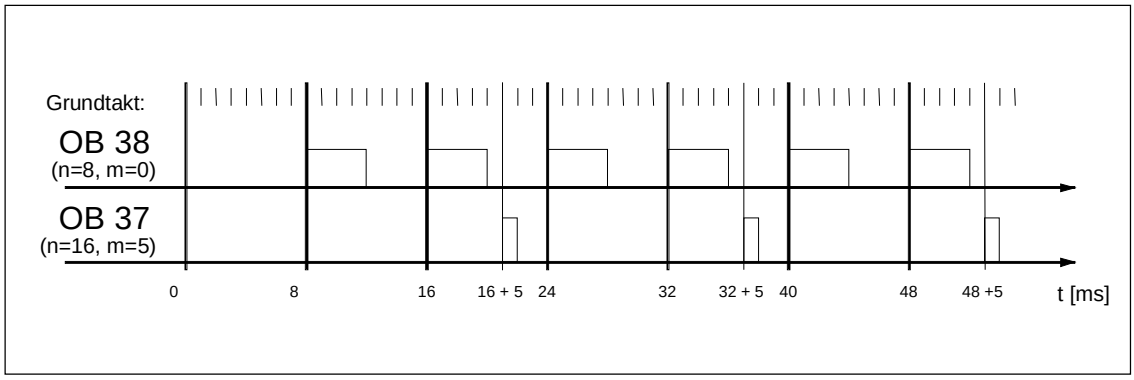
Die neun zur Verfügung stehenden Weckalarm-OBs geben in ihrer Voreinstellung Zeittakte vor (siehe nachfolgende Tabelle). Der Defaultzeittakt wird wirksam, wenn der ihm zugeordnete Weckalarm-OB geladen ist. Sie können jedoch durch Parametrierung die voreingestellten Werte ändern. Die Obergrenze entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten.

Phasenverschiebung bei Weckalarmen

Um zu verhindern, daß die Weckalarme verschiedener Weckalarm-OBs zum gleichen Zeitpunkt eine Startaufforderung erhalten und dadurch möglicherweise ein Zeitfehler (Zykluszeitüberschreitung) entsteht, haben Sie die Möglichkeit, eine Phasenverschiebung vorzugeben. Die Phasenverschiebung sorgt dafür, daß die Bearbeitung eines Weckalarms nach Ablauf des Zeittaktes um einen bestimmten Zeitraum verschoben wird.

Phasenverschiebung = m x Grundtakt (mit $0 \leq m < n$)

Nachfolgendes Bild zeigt die Bearbeitung eines Weckalarm-OBs mit Phasenverschiebung (OB 37) im Gegensatz zu einem Weckalarm ohne Phasenverschiebung (OB 38).



Priorität der Weckalarm-OBs

Nachfolgende Tabelle zeigt die voreingestellten Zeittakte und Prioritätsklassen der Weckalarm-OBs. Zeittakt und Prioritätsklasse können durch Parametrierung geändert werden.

Weckalarm-OBs	Zeittakt in ms	Prioritätsklasse
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

4.2.5.5 Organisationsbausteine für den Prozeßalarm (OB 40 bis OB 47)

Die S7-CPUs stellen Prozeßalarm-OBs zur Verfügung, die auf Signale aus den Baugruppen (z. B. Signalbaugruppen SMs, Kommunikationsprozessoren CPs, Funktionsbaugruppen FMs) reagieren. Mit STEP 7 können Sie für parametrierbare Digital- und Analogbaugruppen einstellen, welches Signal den OB starten soll. Bei CPs und FMs verwenden Sie hierzu die entsprechenden Parametriermasken.

Prozeßalarme werden ausgelöst, wenn eine prozeßalarmfähige Signalbaugruppe mit parametrierter Prozeßalarmfreigabe ein empfangenes Prozeßsignal an die CPU weiterleitet oder eine Funktionsbaugruppe der CPU einen Alarm meldet.

Regeln für Prozeßalarme

Prozeßalarme können nur bearbeitet werden, wenn sich ein entsprechender Organisationsbaustein im CPU-Programm befindet. Ist dies nicht der Fall, wird eine Fehlermeldung in den Diagnosepuffer eingetragen und asynchrone Fehlerbehandlung ausgeführt (siehe "Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)").

Prozeßalarm-OBs, die durch Parametrierung abgewählt wurden, können nicht gestartet werden. Die CPU erkennt einen Programmierfehler und geht in STOP.

Parametrieren prozeßalarmfähiger Signalbaugruppen

Jeder Kanal einer prozeßalarmfähigen Signalbaugruppe kann einen Prozeßalarm auslösen. Deshalb müssen Sie mit STEP 7 in den Parametersätzen prozeßalarmfähiger Signalbaugruppen festlegen:

- wodurch ein Prozeßalarm ausgelöst werden soll
- welcher Prozeßalarm-OB bearbeitet werden soll (die Voreinstellung sieht für die Bearbeitung aller Prozeßalarme den OB 40 vor).

Mit STEP 7 aktivieren Sie die Prozeßalarmgenerierung der Funktionsbaugruppen. Weitere Parameter weisen Sie in den Parametriermasken dieser Funktionsbaugruppen zu.

Priorität der Prozeßalarm-OBs

In der Voreinstellung haben die Prozeßalarm-OBs die Prioritätsklassen 16 bis 23. Die Prioritätsklassen können durch Parametrierung geändert werden.

4.2.5.6 Organisationsbausteine für den Anlauf (OB 100 / OB 101 / OB 102)

Anlaufarten

Man unterscheidet die folgende Anlaufarten:

- Wiederanlauf (nicht bei S7-300 und S7-400H)
- Neustart
- Kaltstart

Der folgenden Tabelle können Sie entnehmen, welcher OB das Betriebssystem im Anlauf jeweils aufruft.

Anlaufart	Zugehöriger OB
Wiederanlauf	OB 101
Neustart	OB 100
Kaltstart	OB 102

Startereignisse für Anlauf-OBs

Die CPU führt einen Anlauf durch

- nach NETZ-EIN
- wenn Sie den Betriebsartenschalter von STOP "RUN"/"RUN-P" umschalten
- nach Anforderung durch eine Kommunikationsfunktion
- nach Synchronisierung im Multicomputing-Betrieb
- bei einem H-System nach dem Ankoppeln (nur auf Reserve-CPU)

Abhängig vom Startereignis, von der vorliegenden CPU und deren eingestellten Parametern wird der zughörige Anlauf-OB (OB 100, OB 101 oder OB 102) aufgerufen.

Anlaufprogramm

Sie können die Randbedingungen für das Anlaufverhalten (Initialisierungswerte für RUN, Anlaufwerte für Peripheriebaugruppen) Ihrer CPU bestimmen, indem Sie Ihr Programm für den Anlauf in den Organisationsbausteinen OB 100 für Neustart, OB 101 für Wiederanlauf oder OB 102 für Kaltstart hinterlegen.

Das Anlaufprogramm kann beliebig lang sein, für seine Ausführung besteht keine zeitliche Begrenzung, die Zykluszeitüberwachung ist nicht aktiv. Zeit- oder alarmgesteuerte Bearbeitung ist im Anlaufprogramm nicht möglich. Im Anlauf führen alle Digitalausgänge den Signalzustand 0.

Anlaufart nach manuellem Anlauf

Bei S7-300-CPUs ist nur manueller Neustart oder Kaltstart (nur CPU 318-2) möglich.

Bei einigen S7-400-CPUs können Sie mit dem Betriebsartenschalter und dem Anlaufartenschalter (CRST/WRST) einen manuellen Wiederanlauf oder Kaltstart durchführen, wenn dies bei der Parametrierung mit STEP 7 so festgelegt wurde. Manueller Neustart ist ohne Parametrierung möglich.

Anlaufart nach automatischem Anlauf

Bei S7-300-CPUs ist nach NETZ-EIN nur Neustart möglich.

Bei S7-400-CPUs können Sie festlegen, ob ein automatischer Anlauf nach NETZ-EIN zu einem Neustart oder einem Wiederanlauf führt.

Löschen des Prozeßabbilds

Beim Wiederanlauf einer S7-400-CPU wird nach der Bearbeitung des Restzyklus defaultmäßig das Prozeßabbild der Ausgänge gelöscht. Sie können das Löschen des Prozeßabbildes abwählen, wenn das Anwenderprogramm nach Wiederanlauf mit den Werten weiterarbeiten soll, die vor dem Wiederanlauf aktuell waren.

Soll-Ist-Baugruppenüberwachung

Sie können durch Parametrierung festlegen, daß vor dem Anlauf überprüft wird, ob alle Baugruppen, die in der Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind und der Baugruppentyp stimmt.

Ist die Baugruppenüberprüfung aktiviert, wird der Anlauf nicht durchgeführt, wenn ein Soll-Ist-Unterschied festgestellt wird.

Überwachungszeiten

Um einen fehlerfreien Anlauf des Automatisierungssystems zu gewährleisten, können Sie die folgenden Überwachungszeiten parametrieren:

- die maximal zulässige Zeit für die Übertragung der Parameter zu den Baugruppen
- die maximal zulässige Zeit für die Fertigmeldung der Baugruppen nach NETZ-EIN
- bei S7-400-CPUs die maximale Unterbrechungszeit in der noch ein Wiederanlauf zulässig ist.

Nach Ablauf der Überwachungszeiten geht die CPU in STOP, bzw. ist nur Neustart möglich.

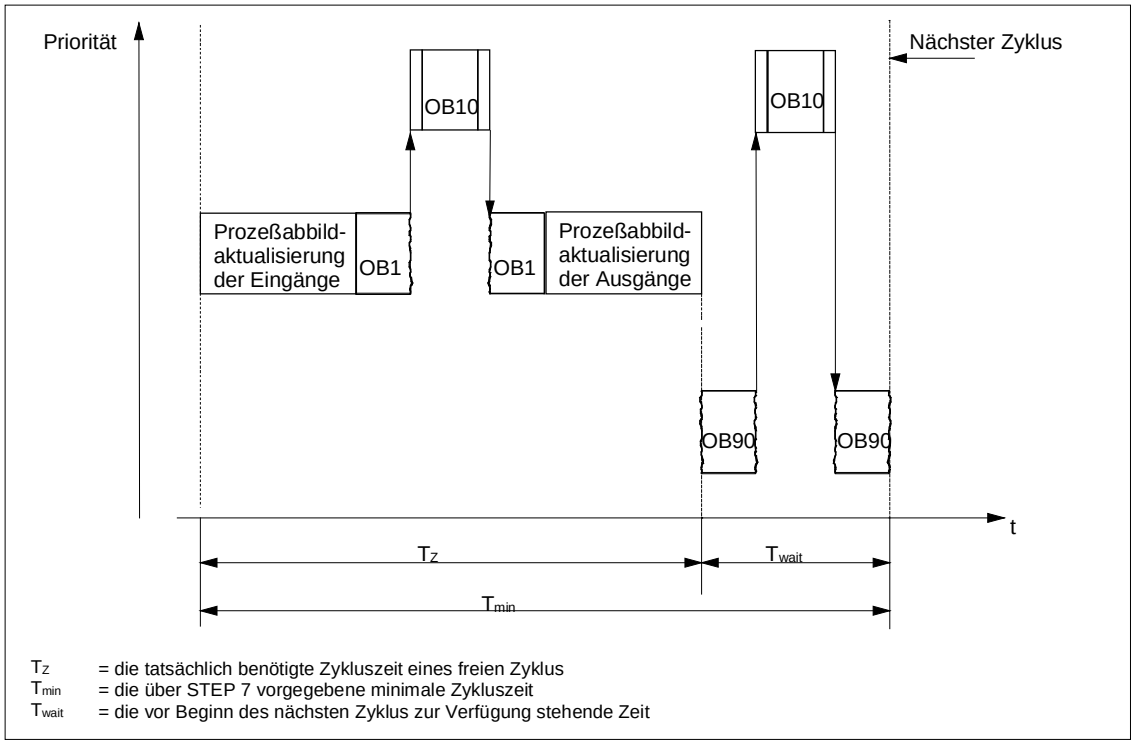
4.2.5.7 Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90)

Falls Sie mit STEP 7 eine minimale Zykluszeit vorgegeben haben und diese größer als die tatsächliche Zykluszeit ist, steht der CPU am Ende des zyklischen Programms Bearbeitungszeit zur Verfügung. Diese wird für die Bearbeitung des Hintergrund-OBs genutzt. Falls der OB 90 auf Ihrer CPU nicht vorhanden ist, wartet die CPU, bis die vorgegebene minimale Zykluszeit abgelaufen ist. Sie können über den OB 90 also zeitunkritische Prozesse ablaufen lassen und damit Wartezeiten vermeiden.

Priorität des Hintergrund-OB

Der Hintergrund-OB hat die Prioritätsklasse 29, was der Priorität 0.29 entspricht. Er ist also der OB mit der niedrigsten Priorität. Die Prioritätsklasse kann durch Parametrierung nicht geändert werden.

Das nachfolgende Bild zeigt ein Beispiel für die Bearbeitung des Hintergrundzyklus, des Freien Zyklus und des OB 10 (bei bisherigen CPUs).



T_Z = die tatsächlich benötigte Zykluszeit eines freien Zyklus
T_{min} = die über STEP 7 vorgegebene minimale Zykluszeit
T_{wait} = die vor Beginn des nächsten Zyklus zur Verfügung stehende Zeit

Programmieren des OB 90

Die Laufzeit des OB 90 wird vom Betriebssystem der CPU nicht überwacht, so daß Sie im OB 90 Schleifen beliebiger Länge programmieren können. Achten Sie auf die Konsistenz der von Ihnen im Hintergrundprogramm verwendeten Daten, indem Sie folgendes bei Ihrer Programmierung berücksichtigen:

- die Rücksetzereignisse des OB 90 (siehe Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen"),
- die zum OB 90 asynchrone Prozeßabbildaktualisierung.

4.2.5.8 Organisationsbausteine für die Fehlerbearbeitung (OB 70 bis OB 87 / OB 121 bis OB 122)

Arten von Fehlern

Die Fehler, die S7-CPUs erkennen und auf die Sie mit Hilfe von Organisationsbausteinen reagieren können, lassen sich in zwei Kategorien einteilen:

- Synchrone Fehler: Diese Fehler können einem bestimmten Teil des Anwenderprogramms zugeordnet werden. Der Fehler wird während der Bearbeitung von einer bestimmten Operation ausgelöst. Ist der entsprechende synchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt.

- Asynchrone Fehler: Diese Fehler lassen sich nicht direkt dem bearbeiteten Anwenderprogramm zuordnen. Es handelt sich um Prioritätsklassenfehler, Fehler im Automatisierungssystem (z. B. Baugruppendefekte) oder um Redundanzfehler. Ist der entsprechende asynchrone Fehler-OB nicht geladen, geht die CPU in STOP, wenn der Fehler auftritt (Ausnahmen: OB 70, OB 72, OB 81).

Nachfolgende Tabelle zeigt die Arten von Fehlern, die auftreten können, aufgeteilt nach der Kategorie der Fehler-OBs.

Asynchrone Fehler / Redundanzfehler	Synchrone Fehler
OB 70 Peripherie-Redundanzfehler (nur H-CPUs)	OB 121 Programmierfehler (z. B. DB ist nicht geladen)
OB 72 CPU-Redundanzfehler (nur in H-CPUs, z. B. Ausfall einer CPU)	OB 122 Peripheriezugriffsfehler (z. B. Zugriff auf eine Signalbaugruppe, die nicht vorhanden ist)
OB 73 Kommunikations-Redundanzfehler (nur in H-CPUs, z. B. Redundanzverlust einer hochverfügbaren S7-Verbindung)	
OB 80 Zeitfehler (z. B. Zykluszeit überschritten)	
OB 81 Stromversorgungsfehler (z. B. Batteriefehler)	
OB 82 Diagnosealarm (z. B. Kurzschluß in der Eingangsbaugruppe)	
OB 83 Ziehen-/Stecken-Alarm (z. B. Ziehen einer Eingabebaugruppe)	
OB 84 CPU-Hardwarefehler (Fehler an der Schnittstelle zum MPI-Netz)	
OB 85 Programmablauffehler (z. B. OB ist nicht geladen)	
OB 86 Baugruppenträgerausfall	
OB 87 Kommunikationsfehler (z. B. falsche Telegrammkennung bei Globaldaten-Kommunikation)	

Verwenden der OBs für synchrone Fehler

Synchrone Fehler werden während der Bearbeitung einer bestimmten Operation verursacht. Wenn diese Fehler auftreten, erstellt das Betriebssystem einen Eintrag im U-Stack und startet den OB für synchrone Fehler.

Die Fehler-OBs, die von synchronen Fehlern aufgerufen werden, werden als Teil des Programms in derselben Prioritätsklasse wie der Baustein bearbeitet, der beim Erkennen des Fehlers bearbeitet wird. OB 121 und OB 122 können also auf die Werte, die zum Zeitpunkt der Unterbrechung in den Akkumulatoren und anderen Registern gespeichert wurden, zugreifen.

Sie können die Werte dazu verwenden, auf die Fehlerbedingung zu reagieren und dann zu der Bearbeitung Ihres Programms zurückzukehren (z. B. bei Zugriffsfehler auf eine Analogeingabebaugruppe im OB 122 mit der SFC 44 RPL_VAL einen Ersatzwert vorgeben). Damit belasten die Lokaldaten der Fehler-OBs aber auch zusätzlich den L-Stack dieser Prioritätsklasse.

Bei S7-400-CPUs kann aus einem synchronen Fehler-OB ein weiterer synchroner Fehler-OB gestartet werden. Bei S7-300 CPUs ist dies nicht möglich.

Verwenden der OBs für asynchrone Fehler

Wenn das Betriebssystem der CPU einen asynchronen Fehler entdeckt, dann startet es den entsprechenden Fehler-OB (OB 70 bis OB 73 und OB 80 bis OB 87). Die OBs für asynchrone Fehler haben die höchste Priorität: Sie können von anderen OBs nicht unterbrochen werden, wenn alle asynchronen Fehler-OBs die gleiche Priorität haben. Treten mehrere asynchrone Fehler-OBs gleicher Priorität gleichzeitig auf, werden sie in der Reihenfolge ihres Auftretens bearbeitet.

Maskieren von Startereignissen

Sie können mit Systemfunktionen (SFC) die Startereignisse für einige Fehler-OBs maskieren bzw. verzögern oder sperren. Nähere Informationen hierzu und zu den Organisationsbausteinen im einzelnen entnehmen Sie dem Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Art des Fehler-OBs	SFC	Funktion der SFC
Synchrone Fehler-OBs	SFC 36 MSK_FLT	Einzelne Synchronfehlerereignisse maskieren. Maskierte Fehlerereignisse starten keinen Fehler-OB und führen nicht zur programmierten Ersatzreaktion.
	SFC 37 DMSK_FLT	Synchronfehlerereignisse demaskieren
Asynchrone Fehler-OBs	SFC 39 DIS_IRT	Alarm- und Asynchronfehlerereignisse pauschal sperren. Gesperrte Fehlerereignisse starten in keinem nachfolgenden CPU-Zyklus Fehler-OBs und führen nicht zur programmierten Ersatzreaktion.
	SFC 40 EN_IRT	Alarm- und Asynchronfehlerereignisse freigeben
	SFC 41 DIS_AIRT	Höherpriore Alarm- und Asynchronfehlerereignisse bis zum OB-Ende verzögern
	SFC 42 EN_AIRT	Höherpriore Alarm- und Asynchronfehlerereignisse freigeben

Hinweis

Um Alarme zu ignorieren, ist es effektiver diese per SFC im Anlauf zu sperren, anstatt einen leeren OB (mit Inhalt BE) zu laden.

5 Starten und Bedienen

5.1 Starten des SIMATIC Managers

5.1.1 Starten von STEP 7

Nach dem Start von Windows finden Sie auf der Windows-Oberfläche ein Symbol für den SIMATIC Manager, dem Einstieg in die STEP 7-Software.

STEP 7 wird am schnellsten mit einem Doppelklick auf das Symbol ”SIMATIC Manager” gestartet. Daraufhin wird das Fenster des SIMATIC Managers geöffnet. Von hier aus lassen sich alle von Ihnen installierten Funktionen sowohl des Basispakets als auch der Optionspakete ansprechen.

Alternativ können Sie den SIMATIC Manager auch über die Schaltfläche ”Start” auf der Task-Leiste in Windows 95/98/NT starten. Den Eintrag finden Sie unter ”Simatic”.

Hinweis

Nähere Informationen über Windows-Standardbedienungen und -Standardoptionen finden Sie in Ihrem Windows-Benutzerhandbuch oder in der Online-Hilfe Ihres Windows Betriebssystems.

SIMATIC Manager

Der SIMATIC Manager ist die Einstiegsoberfläche für die Konfigurierung und Programmierung. Sie können:

- Projekte einrichten
- Hardware konfigurieren und parametrieren
- Hardware-Vernetzungen projektieren
- Bausteine programmieren
- Ihre Programme testen und inbetriebnehmen

Der Zugang zur Funktionalität ist objektorientiert gestaltet und intuitiv und leicht erlernbar.

Sie können mit dem SIMATIC Manager

- offline, d.h. ohne angeschlossene Steuerung oder
- online, d.h. mit angeschlossener Steuerung

arbeiten. (Beachten Sie dabei die entsprechenden Sicherheitshinweise.)

Weiteres Vorgehen

Sie erstellen Automatisierungslösungen in Form von ”Projekten”. Sie erleichtern sich das Arbeiten, wenn Sie sich zuvor mit einigen grundlegenden Themen vertraut machen:

- der Benutzungsoberfläche,
- einigen Basisbedienungen,
- der Online-Hilfe.

5.1.2 Starten von STEP 7 mit vorgegebenen Startparametern

Ab STEP 7 V5 können Sie sich mehrere Symbole des SIMATIC Managers erzeugen und in der Aufrufzeile Startparameter anzugeben. Dadurch können Sie den SIMATIC Manager veranlassen, auf das durch diese Parameter beschriebene Objekt zu positionieren. Damit gelangen Sie einfach per Doppelklick sofort an die entsprechenden Stellen in einem Projekt.

Beim Aufruf von **s7tgtopx.exe** können Sie folgende Startparameter angeben:

/e <vollständiger physikalischer Projektpfad>

/o <logischer Pfad des Objektes, auf welches positioniert werden soll>

/h <ObjektID> /onl oder /off

Die geeigneten Parameter lassen sich am einfachsten wie folgt ermitteln.

Ermitteln der Parameter durch Kopieren und Einfügen

Gehen Sie so vor:

1. Legen Sie auf ihrem Desktop eine neue Verknüpfung mit der Datei s7tgtopx.exe an.
2. Lassen Sie sich das Dialogfeld zu den Eigenschaften anzeigen.
3. Wählen Sie das Register "Verknüpfung". Der Eintrag unter "Ziel" wird nun wie folgt erweitert:
4. Markieren Sie das gewünschte Objekt im SIMATIC Manager.

5. Kopieren Sie das Objekt mit der Tastenkombination CTRL+C in die Zwischenablage.
6. Positionieren Sie den Cursor an das Ende des Eintrags "Ziel" im Register "Verknüpfung".
7. Fügen Sie den Inhalt der Zwischenablage mit der Tastenkombination CTRL+V ein.
8. Schließen Sie das Dialogfeld mit "OK".

Beispiel von Parameterangaben:

/e F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p

/o “1,8:MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1”

/h T00112001;129;T00116001;1;T00116101;16e

Hinweis zum Aufbau des Projektpfads

Der Projektpfad ist der physikalische Pfad im Dateisystem. UNC Notation wird hier nicht unterstützt, also z.B. F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p.

Der vollständige logische Pfad ist folgendermaßen aufgebaut:

[Sichtkennung,Onlinekennung]:Projektname\{Objektname\}*\ Objektname

Beispiel: /o 1,8:MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1

Hinweis zum Aufbau des logischen Pfads

Der vollständige logische Pfad sowie die ObjektID können nur mit Hilfe von Kopieren und Einfügen erstellt werden. Es besteht jedoch auch die Möglichkeit, den vom Benutzer lesbaren Pfad anzugeben, d.h. im o.g. Beispiel:

/o “MyConfig\SIMATIC 400(1)\CPU416-1\S7-Programm(1)\Bausteine\FB1”.

Mit **/onl** bzw. **/off** kann der Benutzer angeben, ob der Pfad im Online oder Offline-Fenster gültig ist. Diese Angabe ist bei der Vorgehensweise mit Kopieren und Einfügen nicht erforderlich.

Wichtig: Falls Leerzeichen im Pfad vorkommen, muß der Pfad zwischen Hochkommas gesetzt werden.

5.1.3 Aufrufen der Hilfefunktionen

Online-Hilfe

Die Online-Hilfe bietet Ihnen Informationen für die Stelle an, an der Sie sie benötigen. So können Sie schnell und zielsicher Information nachschlagen, ohne in Handbüchern suchen zu müssen. In der Online-Hilfe finden Sie:

- **Hilfethemen**: bietet verschiedene Zugänge zum Anzeigen von Hilfeinformation.
- **Hilfe zum Kontext** (Taste F1): zeigt Informationen zum markierten Objekt oder zum aktiven Dialogfeld bzw. Fenster an.
- **Einführung**: gibt einen knappen Überblick über Anwendung, wesentliche Merkmale und Funktionsumfang einer Applikation.
- **Erste Schritte**: faßt erste Handlungen zusammen, die Sie durchführen müssen, um zu einem ersten Erfolg zu kommen.
- **Hilfe benutzen**: bietet eine Beschreibung der Möglichkeiten, die Ihnen zur Verfügung stehen, um bestimmte Informationen in der Hilfe zu finden.
- **Info**: liefert Informationen zur aktuellen Version der Applikation.

Über das Menü "Hilfe" können Sie von jedem Fenster aus auf Themen zugreifen, die sich auf die aktuelle Dialogsituation beziehen.

Aufrufen der Online-Hilfe

Sie können die Online-Hilfe auf verschiedene Arten aufrufen:

- Wählen Sie einen Menübefehl aus dem Menü "Hilfe" in der Menüleiste.
- In einem Dialogfeld klicken Sie auf die Schaltfläche ”Hilfe”. Sie erhalten dann Hilfe zu diesem Dialogfeld.
- Positionieren Sie den Mauszeiger in einem Fenster oder Dialogfeld auf das Thema, zu dem Sie Hilfe benötigen, und drücken Sie die Taste F1 oder wählen Sie den Menübefehl **Hilfe > Hilfe zum Kontext**.
- Fragezeichen-Cursor von Windows.

Die drei letztgenannten Arten werden als kontextabhängige Hilfe bezeichnet.

Aufrufen der Kurzhilfe

Eine Kurzhilfe zu Schaltflächen in der Funktionsleiste wird angezeigt, wenn Sie den Cursor auf eine Schaltfläche positionieren und ihn dort kurze Zeit stehen lassen.

Schriftgröße ändern

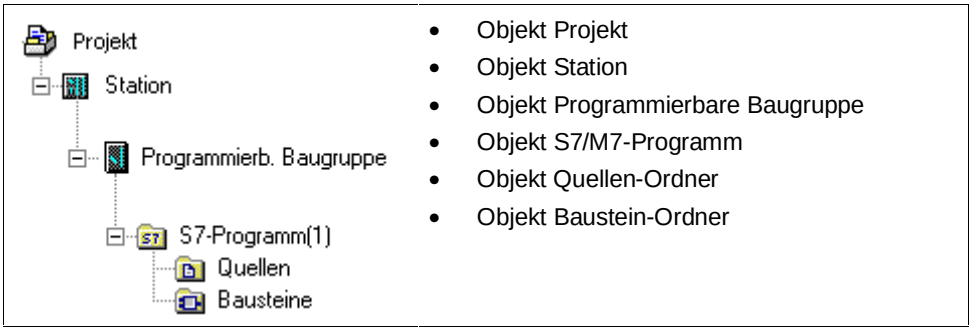
Mit dem Menübefehl **Optionen > Schriftart** im Hilfefenster können Sie die Schriftgröße auf ”Klein”, ”Standard” oder ”Groß” stellen.

5.2 Objekte und Objekthierarchie

5.2.1 Objekte und Objekt-Hierarchie

Vergleichbar zur Verzeichnisstruktur des Windows-Explorers mit Ordnern und Dateien, ist auch im SIMATIC Manager die Objekt-Hierarchie für Projekte und Bibliotheken aufgebaut.

Nachfolgendes Bild zeigt ein Beispiel für eine Objekt-Hierarchie.



Objekte dienen

- als Träger von Eigenschaften,
- als Ordner,
- als Träger von Funktionen (z. B. zum Start einer bestimmten Applikation).

Objekte als Träger von Eigenschaften

Objekte können sowohl Funktionen als auch Eigenschaften (z. B. Einstellungen) tragen. Nach dem Selektieren eines Objekts können Sie

- mit dem Menübefehl **Bearbeiten > Objekt öffnen** das Objekt bearbeiten.
- mit dem Menübefehl **Bearbeiten > Objekteigenschaften** ein Dialogfeld aufblenden und darin objektspezifische Einstellungen vornehmen.

Auch ein Ordner kann Träger von Eigenschaften sein.

Objekte als Ordner

In einem Ordner können sich weitere Ordner oder Objekte befinden. Diese werden angezeigt, wenn Sie den Ordner öffnen.

Objekte als Träger von Funktionen

Wenn Sie ein Objekt öffnen, wird ein Fenster aufgeblendet, in dem Sie das Objekt bearbeiten können.

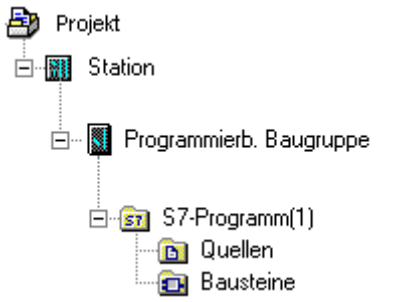
Ein Objekt ist entweder Ordner oder Träger von Funktionen. Eine Ausnahme bilden die Stationen: Sie sind sowohl Ordner (für programmierbare Baugruppen) als auch Träger von Funktionen (zum Konfigurieren der Hardware).

- Bei Doppelklick auf eine Station werden die darin enthaltenen Objekte angezeigt: die programmierbaren Baugruppen und die Stationskonfiguration (Station als Ordner).
- Nach dem Öffnen einer Station mit dem Menübefehl **Bearbeiten > Objekt öffnen** können Sie diese Station konfigurieren und parametrieren (Station als Träger einer Funktion). Der Menübefehl hat die gleiche Auswirkung wie ein Doppelklick auf das Objekt Hardware.

5.2.2 Objekt Projekt

Das Projekt repräsentiert die Gesamtheit aller Daten und Programme einer Automatisierungslösung und steht an der Spitze einer Objekthierarchie

Plazierung in der Projektansicht



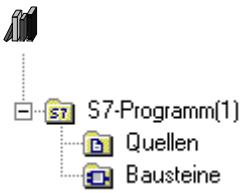
- **Objekt Projekt**
- Objekt Station
- Objekt Programmierbare Baugruppe
- Objekt S7/M7-Programm
- Objekt Quellen-Ordner
- Objekt Baustein-Ordner

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Projekt	• Anlegen eines Projekts • Archivieren von Projekten und Bibliotheken • Drucken einer Projektdokumentation • Reorganisation • Bedienerrelevante Texte übersetzen und bearbeiten • Einfügen von OS-Objekten • Bearbeiten von Projekten durch mehrere Personen • Umsetzen von Version 1-Projekten • Umsetzen von Version 2-Projekten • Einstellen der PG/PC-Schnittstelle

Symbol	Objekte in der Projektebene	Auswahl wichtiger Funktionen
	Station: SIMATIC 300-Station SIMATIC 400-Station	• Einfügen von Stationen • Stationen sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Stationsebene). Weitere Funktionen finden Sie bei Objekt Station
	S7-Programm M7-Programm	• S7/M7-Programm ohne Station und CPU • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm
	Netz zum Start der Applikation für die Netzprojektierung und zur Einstellung von Netzeigenschaften	• Eigenschaften von Subnetzen und Kommunikationsteilnehmern • Übersicht: Globaldaten-Kommunikation • Vorgehensweise zur Konfigurierung der GD-Kommunikation

5.2.3 Objekt Bibliothek

Eine Bibliothek kann S7-Programme/M7-Programme enthalten und dient zur Ablage von Bausteinen. Sie steht an der Spitze einer Objekthierarchie.



- **Objekt Bibliothek**
- Objekt S7/M7-Programm
- Objekt Quellen-Ordner
- Objekt Baustein-Ordner

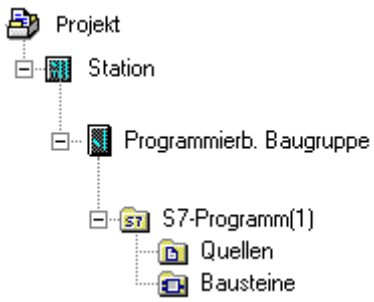
Symbol	Objektordner	Auswahl wichtiger Funktionen
	Bibliothek	• Übersicht der Standardbibliotheken • Arbeiten mit Bibliotheken • Archivieren von Projekten und Bibliotheken

Symbol	Objekte in der Bibliotheksebene	Auswahl wichtiger Funktionen
	S7-Programm M7-Programm	• Einfügen eines S7-/M7-Programms • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm

5.2.4 Objekt Station

Eine SIMATIC 300/400-Station repräsentiert einen S7-Hardware-Aufbau mit einer oder mehreren programmierbaren Baugruppen.

Plazierung in der Projektansicht



- Objekt Projekt
- **Objekt Station**
- Objekt Programmierbare Baugruppe
- Objekt S7/M7-Programm
- Objekt Quellen Ordner
- Objekt Baustein-Ordner

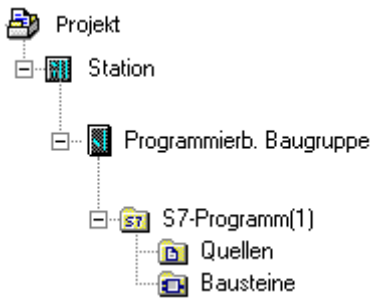
Symbol	Objektordner	Auswahl wichtiger Funktionen
	Station	• Einfügen von Stationen • Station laden in PG • Laden einer Konfiguration in ein Zielsystem • Zurückladen einer Konfiguration aus einer Station • CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen • 'Melden von Systemfehlern' projektieren • Diagnose der Hardware und Anzeigen des Baugruppenzustands • Anzeigen und Ändern des Betriebszustands • Anzeigen und Einstellen von Uhrzeit und Datum • Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU
	SIMATIC PC-Station	• Erzeugen und parametrieren von SIMATIC PC-Stationen • Verbindungen projektieren für eine SIMATIC PC-Station

Symbol	Objekte in der Stationsebene	Auswahl wichtiger Funktionen
	Hardware	• Grundsätzliche Bedienung beim Hardware konfigurieren • Grundsätzliche Schritte beim Konfigurieren einer Station • Überblick: Vorgehensweise beim Konfigurieren und Parametrieren eines zentralen Aufbaus • Grundsätzliche Vorgehensweise zum Konfigurieren eines DP-Mastersystems • Konfigurieren des Multicomputing-Betriebs
	Programmierbare Baugruppe	• Programmierbare Baugruppen sind gleichzeitig Objekte (Stationsebene) und Objekt-Ordner (Ebene "Programmierbare Baugruppen). Weitere Funktionen finden Sie bei Objekt Programmierbare Baugruppe

5.2.5 Objekt Programmierbare Baugruppe

Eine programmierbare Baugruppe repräsentiert die Parametrierungsdaten einer programmierbaren Baugruppe (CPUxxx, FMxxx, CPxxx). Die Systemdaten von Baugruppen, die keinen remanenten Speicher haben (z. B. CP441), werden über die CPU der Station geladen. Solchen Baugruppen ist deshalb kein Objekt ”Systemdaten” zugeordnet und sie werden nicht in der Projekthierarchie angezeigt.

Plazierung in der Projektansicht



- Objekt Projekt
- Objekt Station
- **Objekt Programmierbare Baugruppe**
- Objekt S7/M7-Programm
- Objekt Quellen Ordner
- Objekt Baustein-Ordner

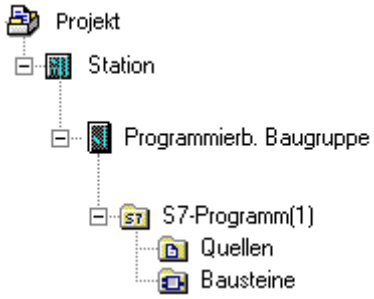
Symbol	Objektordner	Auswahl wichtiger Funktionen
	Programmierbare Baugruppe	• Überblick: Vorgehensweise beim Konfigurieren und Parametrieren eines zentralen Aufbaus • CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen • 'Melden von Systemfehlern' projektieren • Diagnose der Hardware und Anzeigen des Baugruppenzustands • Laden über EPROM-Memory Cards • Paßwortschutz für Zugriff auf Zielsysteme • Forcefenster anzeigen • Anzeigen und Ändern des Betriebszustands • Anzeigen und Einstellen von Uhrzeit und Datum • Einstellen des Betriebsverhaltens • Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU • Diagnosesymbole in der Online-Sicht • Aufteilung der Speicherbereiche • Speichern geladener Bausteine auf integriertem EPROM • Aktualisieren des Betriebssystems auf dem Zielsystem

Symbol	Objekte in der Ebene "Programmierbare Baugruppen"	Auswahl wichtiger Funktionen
	Programme: S7-Programm M7-Programm Programm	• Einfügen eines S7-/M7-Programms • S7-/M7-Programme sind gleichzeitig Objekte (Projektebene) und Objekt-Ordner (Programmebene). Weitere Funktionen finden Sie bei Objekt S7/M7-Programm
	Verbindungen zur Definition von Verbindungen im Netzwerk	• Vernetzen von Stationen innerhalb eines Projekts • Verbindungstypen und Verbindungspartner • Wissenswertes zu den verschiedenen Verbindungstypen • Eingeben einer neuen Verbindung • Verbindungen projektieren für Baugruppen einer SIMATIC Station

5.2.6 Objekt S7/M7-Programm

Ein (S7-/M7-) Programm ist ein Ordner für Software zu S7/M7-CPU-Baugruppen oder für Software zu Nicht-CPU-Baugruppen (z. B. programmierbare CP- oder FM-Baugruppen).

Plazierung in der Projektansicht



- Objekt Projekt
- Objekt Station
- Objekt Programmierbare Baugruppe
- **Objekt S7/M7-Programm**
- Objekt Quellen Ordner
- Objekt Baustein-Ordner

Symbol	Objektordner	Auswahl wichtiger Funktionen
	S7-Programm	• Einfügen eines S7-/M7-Programms • Einstellung des Operandenvorrangs • Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen • Vergeben von Meldenummern • Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten • Bedienerrelevante Texte übersetzen und bearbeiten • CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen • Maßnahmen im Programm zur Störungsbehandlung
	M7-Programm	• Vorgehensweise für M7-Systeme
	Programm	• Erstellen der Software im Projekt (prinzipiell)

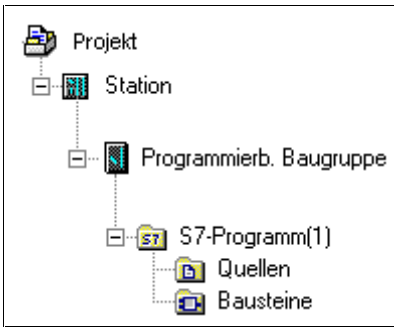
Symbol	Objekte in der Programmebene	Auswahl wichtiger Funktionen
	Symboltabelle zur Zuordnung von Symbolen zu Signalen und anderen Variablen	• Absolute und symbolische Adressierung • Struktur und Bestandteile der Symboltabelle • Eingabemöglichkeiten von globalen Symbolen • Allgemeine Hinweise zur Eingabe von Symbolen • Symbolbezogene Meldungen zuordnen und bearbeiten • Bedienerrelevante Texte übersetzen und bearbeiten • Projektieren von BuB-Attributen über die Symboltabelle • Bearbeiten des Kommunikationsattributs • Exportieren und Importieren von Symboltabellen
	Quellen-Ordner	• Weitere Funktionen finden Sie bei Objekt Quellen-Ordner
	Baustein-Ordner	• Weitere Funktionen finden Sie bei Objekt Baustein-Ordner

5.2.7 Objekt Baustein-Ordner

Ein Baustein-Ordner einer Offline-Ansicht kann enthalten: Codebausteine (OB, FB, FC, SFB, SFC), Datenbausteine (DB), anwenderdefinierte Datentypen (UDT) und Variablentabellen. Das Objekt Systemdaten repräsentiert Systemdatenbausteine.

Der Baustein-Ordner einer Online-Ansicht enthält die ausführbaren Programmteile, die resident im Zielsystem geladen sind.

Plazierung in der Projektansicht



- Objekt Projekt
- Objekt Station
- Objekt Programmierbare Baugruppe
- Objekt S7/M7-Programm
- Objekt Quellen Ordner
- **Objekt Baustein-Ordner**

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Bausteine	• Laden mit Projektverwaltung • Laden ohne Projektverwaltung • Übersicht der möglichen Referenzdaten • Umverdrahten • Bausteine vergleichen • Bedienerrelevante Texte übersetzen und bearbeiten • Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute

Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	Bausteine allgemein	• Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen • Anlegen von Bausteinen • Grundlagen zum Programmieren in AWL-Quellen • Bausteine vergleichen
	OB (Organisationsbaustein)	Zusätzliche Funktionen: • Einführung zu Daten- und Parametertypen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Baustein-Hilfen
	FC (Funktionen)	Zusätzliche Funktionen: • Einführung zu Daten- und Parametertypen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Attribute für Bausteine und Parameter

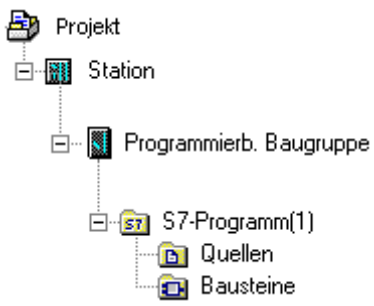
Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	FB (Funktionsbausteine)	Zusätzliche Funktionen: • Einführung zu Daten- und Parametertypen • Verwendung von Multiinstanzen • Laden • Testen mit Programmstatus • Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte • Umverdrahten • Attribute für Bausteine und Parameter • Bausteinbezogenen Meldungen zuordnen und bearbeiten • PCS7-Meldungsprojektierung • Bedienerrelevante Texte übersetzen und bearbeiten • FB-Parametern BuB-Attribute zuweisen
	UDT (Anwenderdefinierte Datentypen)	• Anlegen von Bausteinen • Grundlagen zum Programmieren in AWL- Quellen • Einführung zu Daten- und Parametertypen • Verwenden von anwenderdefinierten Datentypen für den Datenzugriff • Attribute für Bausteine und Parameter
	DB (Globale Datenbausteine)	• Datensicht von Datenbausteinen • Deklarationssicht von Datenbausteinen • Laden • Programmstatus von Datenbausteinen • Einführung zu Daten- und Parametertypen • Verwendung von Multiinstanzen • Attribute für Bausteine und Parameter • Bausteinbezogenen Meldungen zuordnen und bearbeiten (nur Instanz-DBs) • PCS7-Meldungsprojektierung (nur Instanz-DBs) • Bedienerrelevante Texte übersetzen und bearbeiten (nur Instanz-DBs)
	SFC (Systemfunktionen)	• Laden • Attribute für Bausteine und Parameter • Baustein-Hilfen

Symbol	Objekte im Bausteinordner	Auswahl wichtiger Funktionen
	SFB (Systemfunktionsbaustein)	• Laden • Attribute für Bausteine und Parameter • Bausteinbezogenen Meldungen zuordnen und bearbeiten • PCS7-Meldungsprojektierung • Bedienerrelevante Texte übersetzen und bearbeiten • Baustein-Hilfen
	VAT (Variablentabelle)	• Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablentabellen • Einführung zum Testen mit der Variablentabelle • Einführung zum Beobachten von Variablen • Einführung zum Steuern von Variablen • Einführung zum Forcen von Variablen
	Systemdaten (SDB)	SDBs werden nur indirekt bearbeitet über Funktionen aus: • Einführung zum Konfigurieren der Hardware • Eigenschaften von Subnetzen und Kommunikationsteilnehmern • Übersicht: Globaldaten-Kommunikation • Symbolbezogene Meldungen zuordnen und bearbeiten • Laden

5.2.8 Objekt Quellen-Ordner

Ein Quellen-Ordner enthält Quellprogramme in Textform.

Plazierung in der Projektansicht



- Objekt Projekt
- Objekt Station
- Objekt Programmierbare Baugruppe
- Objekt S7/M7-Programm
- **Objekt Quellen-Ordner**
- Objekt Baustein-Ordner

Symbol	Objektordner	Auswahl wichtiger Funktionen
	Quellen-Ordner	• Grundlagen zum Programmieren in AWL-Quellen • Exportieren von Quellen • Importieren von Quellen

Symbol	Objekte im Quellen-Ordner	Auswahl wichtiger Funktionen
	Quelle (z. B. AWL-Quelle)	• Grundlagen zum Programmieren in AWL-Quellen • Anlegen von AWL-Quellen • Einfügen von Bausteinvorlagen in AWL-Quellen • Einfügen des Quellcodes vorhandener Bausteine in AWL-Quellen • Prüfen der Konsistenz in AWL-Quellen • Übersetzen von AWL-Quellen • Generieren von AWL-Quellen aus Bausteinen • Exportieren von Quellen • Importieren von Quellen
	Netzwerkvorlage	• Netzwerkvorlage erstellen • Netzwerkvorlage in ein Programm einfügen

5.2.9 S7/M7-Programm ohne Station und CPU

Sie können Programme erstellen, ohne zuvor SIMATIC-Station konfiguriert zu haben. Damit sind Sie zunächst unabhängig von der zu programmierenden Baugruppe und deren Einstellungen.

Anlegen des S7/M7-Programms

1. Öffnen Sie das entsprechende Projekt über den Menübefehl **Datei > Öffnen** bzw. aktivieren Sie das Projektfenster.
2. Markieren Sie das Projekt im Projektfenster der Ansicht offline.
3. Je nachdem, für welches Zielsystem das erstellte Programm bestimmt ist, wählen Sie den entsprechenden Menübefehl:

   Einfügen > Programm > S7-Programm, wenn Ihr Programm später auf einer SIMATIC S7 laufen soll.

   Einfügen > Programm > M7-Programm, wenn Ihr Programm später auf einer SIMATIC M7 laufen soll.

Der S7/M/-Programm wird eingefügt und im Projektfenster direkt unter dem Projekt angeordnet. Es enthält einen Ordner für die Bausteine und eine leere Symboltabelle. Nun können Sie Bausteine anlegen und programmieren.

Zuordnung zu einer programmierbaren Baugruppe

Baugruppenunabhängig eingefügte Programme können Sie später im Projektfenster einfach einer Baugruppe zuordnen, indem Sie diese Programme auf das Baugruppen-Symbol per Drag&Drop kopieren oder verschieben.

Übernahme in Bibliotheken

Wenn das Programm für das Zielsystem SIMATIC S7 bestimmt ist und gleich einem "Software-Pool" mehrfach verwendet werden soll, kann es auch unter einer Bibliothek eingefügt werden. Zum Testen müssen Programme aber direkt unter einem Projekt liegen, da Sie nur dann eine Verbindung zum Zielsystem herstellen können.

Zugriff auf ein Zielsystem

Wählen Sie die Ansicht online des Projekts. Im Dialogfeld zu den Eigenschaften des Programms können Sie die Adreßeinstellungen vornehmen.

Hinweis

Beim Löschen von Stationen oder programmierbaren Baugruppen werden Sie gefragt, ob auch das enthaltene Programm gelöscht werden soll. Wenn Sie diese Frage verneinen, wird das Programm direkt unter dem Projekt als Programm ohne Station eingehängt.

5.3 Benutzungsoberfläche und Bedienung

5.3.1 Bedienphilosophie

Ziel: einfache Bedienung

Die grafische Oberfläche soll Ihnen eine möglichst intuitive Bedienung ermöglichen. Sie finden deshalb Objekte vor, die Sie aus Ihrer täglichen Arbeitswelt kennen, z. B. Stationen, Baugruppen, Programme, Bausteine.

Die Handlungen, die Sie beim Arbeiten mit STEP 7 durchführen, umfassen die Erzeugung, Auswahl und Manipulation solcher Objekte.

Unterschiede zur tool-orientierten Bedienung

Bei der herkömmlichen tool-orientierten Bedienung mußte man sich zunächst überlegen, welches Tool zum Lösen einer bestimmten Aufgabe erforderlich ist und dann das Tool aufrufen.

Das prinzipielle Vorgehen bei der objektorientierten Bedienphilosophie ist, sich zu überlegen, welches Objekt bearbeitet werden muß und dann das Objekt zu öffnen und zu bearbeiten.

Bei der objektorientierten Bedienphilosophie ist die Kenntnis einer speziellen Befehlssyntax nicht mehr erforderlich. Objekte werden an der Oberfläche durch grafische Symbole repräsentiert, die sich durch Menübefehle oder Mausklicks öffnen lassen.

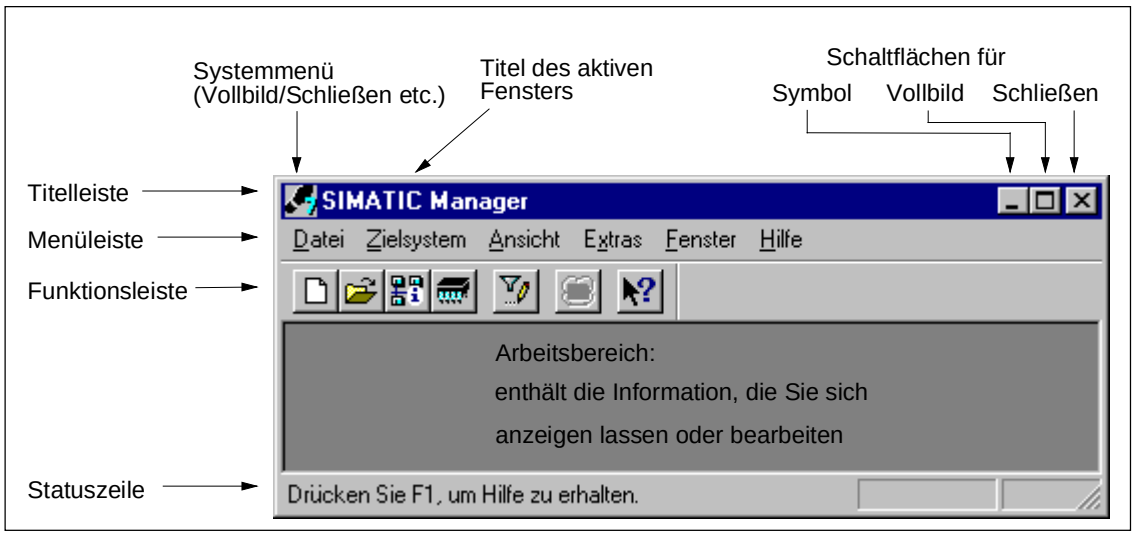
Beim Öffnen eines Objekts wird automatisch die geeignete Softwarekomponente aufgerufen, um den Objektinhalt anzuzeigen oder zu bearbeiten.

Weiterlesen ...

Nachfolgend sind die grundlegende Aktionen beim Bearbeiten von Objekten beschrieben. Machen Sie sich bitte damit vertraut, da auf diese immer wiederkehrenden Grundbedienungen später aufgebaut wird.

5.3.2 Aufbau des Fensters

Die Standardkomponenten eines Fensters sind im nachfolgenden Bild dargestellt:



Titelleiste und Menüleiste

Titelleiste und die Menüleiste befinden sich immer am oberen Rand des Fensters. Die Titelleiste enthält den Fenstertitel und Symbole für die Fenstersteuerung. Die Menüleiste enthält alle Menüs, die im Fenster zur Verfügung stehen.

Funktionsleiste

Die Funktionsleiste enthält Symbole, über die Sie schnell durch Mausklick auf häufig verwendete und aktuell verfügbare Menübefehle zugreifen können. Eine Kurzinformation über die Funktion eines Symbols wird aufgeblendet, wenn Sie den Cursor kurze Zeit auf dem Symbol positioniert lassen, eine zusätzliche Information wird in der Statuszeile angezeigt.

Wenn in der aktuellen Konfiguration kein Zugriff auf ein Symbol möglich ist, so wird dieses grau dargestellt.

Statuszeile

In der Statuszeile werden kontextabhängige Informationen angezeigt.

5.3.3 Elemente in Dialogfeldern

Eingabe in Dialogfelder

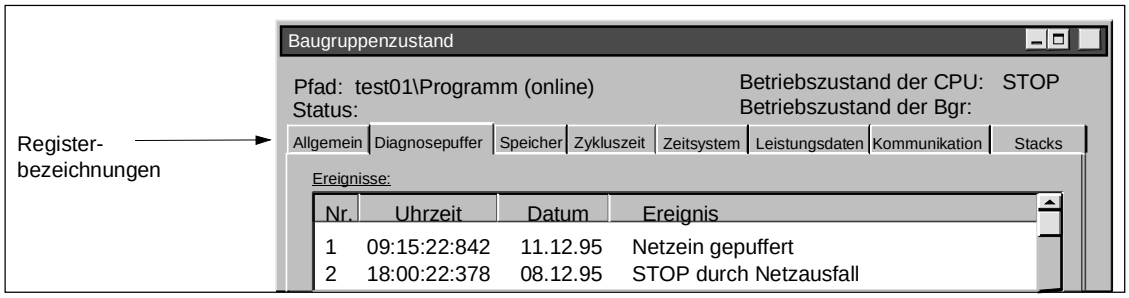
In Dialogfeldern können Sie Informationen eingeben, die für das Ausführen einer bestimmten Aufgabe benötigt werden. Die häufigsten Komponenten von Dialogfeldern sind anhand eines Beispiels im nachfolgenden Bild erklärt:

Listenfelder / Kombinationsfelder

Manchmal sind Textfelder mit einem nach unten gerichteten Pfeil versehen. Dieser Pfeil zeigt an, daß für dieses Feld weitere Auswahlmöglichkeiten zur Verfügung stehen. Klicken Sie auf den Pfeil, um ein Listen- oder Kombinationsfeld aufzuschlagen. Wenn Sie darin einen Eintrag anklicken, wird er automatisch in das Textfeld übernommen.

Registerdialoge

Der Inhalt mancher Dialogfelder ist wegen der besseren Übersicht in Registern organisiert (siehe nachfolgendes Bild).



Die Bezeichnungen der einzelnen Register befinden sich am oberen Rand des Dialogfelds. Um ein bestimmtes Register ”nach vorne” zu holen, klicken Sie einfach auf die Registerbezeichnung.

5.3.4 Erzeugen und Manipulieren von Objekten

Einige grundlegende Handlungen an Objekten sind für alle Objekte gleich. Diese prinzipiellen Handlungsfolgen finden Sie hier zusammenfassend dargestellt. Die Kenntnis wird in den nachfolgenden Kapiteln des Handbuchs bei der Beschreibung von Vorgehensweisen als bekannt vorausgesetzt.

Die übliche Handlungsfolge beim Umgang mit Objekten ist:

- Objekt erzeugen,
- Objekt auswählen,
- Aktionen mit dem Objekt durchführen (z. B. kopieren, löschen).

Ablagepfad für neue Projekte/Bibliotheken einstellen

Vor dem erstmaligen Anlegen neuer Projekte oder Bibliotheken sollten Sie den Ablagepfad für diese Objekte einstellen. Wählen Sie dazu den Menübefehl **Extras > Einstellungen**. Im Register ”Allgemein” des angezeigten Dialogfelds können Sie dann einen Pfadnamen angeben, unter dem neu erzeugte Projekte oder Bibliotheken abgelegt werden.

Erzeugen von Objekten

Der STEP 7-Assistent ”Neues Projekt” unterstützt Sie beim Anlegen eines neuen Projektes und Einfügen von Objekten. Zum Aufruf wählen Sie den Menübefehl **Datei > Assistent 'Neues Projekt'**. In den angezeigten Dialogfeldern können Sie den Aufbau ihres Projektes festlegen und anschließend das Projekt vom Assistenten erzeugen lassen.

Wenn Sie den Assistenten nicht benutzen möchten, können Sie mit dem Menübefehl **Datei > Neu** Projekte und Bibliotheken erzeugen. Diese Objekte bilden die Spitze einer Objekthierarchie. Alle weiteren Objekte der Hierarchie - soweit sie nicht automatisch angelegt werden - können Sie über Befehle im Menü "Einfügen" erzeugen. Eine Ausnahme sind die Baugruppen einer SIMATIC-Station, die nur im Rahmen der Hardware-Konfigurierung und vom Assistenten 'Neues Projekt' angelegt werden.

Öffnen von Objekten

Es gibt mehrere Möglichkeiten, um ein Objekt in der Detailansicht zu öffnen:

- Doppelklick auf das Objektsymbol
- Objekt selektieren und Menübefehl **Bearbeiten > Objekt öffnen**. Dies funktioniert nur bei Objekten, die keine Ordner sind.

Nach dem Öffnen eines Objekts können Sie dessen Inhalt erstellen oder ändern.

Wenn Sie ein Objekt des zuletzt genannten Typs öffnen, wird dessen Inhalt durch eine geeignete Softwarekomponente in einem neuen Fenster zur Bearbeitung dargestellt. Objekte, deren Inhalt bereits anderweitig benutzt wird, können Sie nicht ändern.

Hinweis

Ausnahme: Stationen erscheinen als Ordner für programmierbare Baugruppen (bei Doppelklick) und für die Stationskonfiguration. Wenn Sie auf das Objekt "Hardware" doppelklicken, wird die Applikation für das Konfigurieren der Hardware gestartet. Die gleiche Wirkung hat das Markieren der Station und der anschließende Menübefehl **Bearbeiten > Objekt öffnen**.

Aufbauen einer Objekt-Hierarchie

Lassen Sie sich die Objekthierarchie vom Assistenten 'Neues Projekt' aufbauen. Nach dem Öffnen eines Ordners werden die bereits darin enthaltenen Objekte am Bildschirm dargestellt. Über das Menü "Einfügen" können Sie nun weitere Unterobjekte anlegen, beispielsweise weitere Stationen in einem Projekt. In diesem Menü sind nur die Befehle zum Einfügen solcher Objekte aufrufbar, die im aktuellen Ordner zulässig sind.

Einstellen von Objekteigenschaften

Objekteigenschaften sind Daten des Objekts, die dessen Verhalten festlegen. Der Dialog zur Einstellung von Objekteigenschaften erscheint automatisch, wenn Sie ein Objekt neu erzeugen und dabei Eigenschaften festgelegt werden müssen. Jedoch lassen sich die Eigenschaften auch nachträglich ändern:

Mit dem Menübefehl **Bearbeiten > Objekteigenschaften** wird ein Dialogfeld aufgerufen, in dem sich die Eigenschaften des ausgewählten Objekts ablesen oder einstellen lassen.

Mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften** können Sie Dialogfelder aufrufen und Daten eingeben, die für das Bedienen und Beobachten sowie für die Meldungsprojektierung erforderlich sind.

Um beispielsweise die speziellen Objekteigenschaften eines Bausteins für Bedienen und Beobachten aufrufen zu können, muß der Baustein als B&B-relevant gekennzeichnet sein, d.h. im Register ”Attribute” der Bausteineigenschaften muß das Systemattribut ”S7_m_c” mit dem Wert ”true” eingetragen sein.

Hinweis

- Eigenschaften des Ordners "Systemdaten” und des Objekts "Hardware” lassen sich nicht anzeigen und verändern.
- In den Dialogfeldern zu den Eigenschaften von Objekten eines schreibgeschützten Projektes ist kein Schreiben möglich. In diesem Fall sind die Eingabefelder kontrastvermindert dargestellt.
- Wenn Sie sich die Eigenschaften von programmierbaren Baugruppen anzeigen lassen, dann können Sie die angezeigten Parameter aus Konsistenzgründen nicht bearbeiten. Zum Bearbeiten der Parameter müssen Sie die Applikation "Hardware konfigurieren” aufrufen.
- Wenn Sie Einstellungen an Objekten auf dem Erstellsystem ändern (z. B. Parametrierungsdaten einer Baugruppe), so sind diese zunächst noch nicht auf dem Zielsystem wirksam. Denn dazu müssen sich die Systemdatenbausteine, in denen diese Einstellungen abgespeichert werden, auf dem Zielsystem befinden.
- Wenn Sie ein komplettes Anwenderprogramm laden, werden automatisch die Systemdatenbausteine mit übertragen. Wenn Sie nach dem Laden eines Programms Änderungen an den Einstellungen vornehmen, so können Sie das Objekt ”Systemdaten” nachladen, um die Einstellungen auf das Zielsystem zu bringen.
- Es wird dringend empfohlen die Ordner nur mit STEP 7 zu bearbeiten, da sie physikalisch anders strukturiert sein können, als Sie es im SIMATIC Manager sehen.

Ausschneiden, Einfügen oder Kopieren

Die meisten Objekte können Sie wie unter Windows üblich ausschneiden, einfügen oder kopieren. Die zugehörigen Menübefehle befinden sich im Menü "Bearbeiten".

Sie können auch durch ”Ziehen und Loslassen” (Drag&Drop) Objekte kopieren. Wenn Sie dabei auf ein unzulässiges Ziel zeigen, wird der Cursor als Verbotszeichen dargestellt.

Wenn Sie ein Objekt kopieren, wird damit die gesamte darunter liegende Hierarchie mit kopiert. Dadurch ergeben sich weitreichende Möglichkeiten zur Wiederverwendung von einmal entwickelten Komponenten einer Automatisierungslösung.

Hinweis

Die Verbindungstabelle im Ordner “Verbindungen” können Sie nicht kopieren.Beachten Sie, daß beim Kopieren von Listen bedienerrelevanter Texte nur die Sprachen übernommen werden, die im Zielobjekt installiert sind.

Eine Schritt-für-Schritt-Anleitung zum Kopieren finden Sie in Kopieren von Objekten.

Umbenennen von Objekten

Der SIMATIC Manager vergibt an neu eingefügte Objekte standardisierte Namen. Diese Namen ergeben sich in der Regel aus dem Typ des Objekts und (wenn mehrere Objekte dieses Typs im gleichen Ordner angelegt werden können) einer laufenden Nummer.

So heißt z.B. das erste S7-Programm "S7-Programm(1)", das zweite "S7-Programm(2)". Die Symboltabelle hingegen heißt einfach "Symbole", da sie in jedem übergeordneten Ordner nur einmal existiert.

Die Namen der meisten Objekte (auch eines Projekts) können Sie ändern und Ihnen dadurch eine selbsterklärende Bezeichnung geben.

Bei Projekten dürfen die im Pfad enthaltenen Verzeichnisnamen nicht länger als 8 Zeichen sein. Ansonsten kann es zu Problemen beim Archivieren und beim Einsatz von "C für M7" (Borland-Compiler) kommen.

Änderbare Objektnamen können Sie direkt editieren oder über die Objekteigenschaften ändern.

- Direktes Editieren:
- Wenn Sie auf den Namen eines markierten Objekts zweimal langsam klicken, erscheint ein Rahmen um den Text. Sie können nun die Bezeichnung über die Tastatur editieren.
- Ändern über Objekteigenschaften:
- Wählen Sie dazu das gewünschte Objekt aus, und geben Sie den Menübefehl **Bearbeiten > Objekteigenschaften**. Ändern Sie den Namen im Dialogfeld. Nach dem Schließen des Eigenschaftendialogs ist das Objekt umbenannt und wird mit dem neuen Namen angezeigt.

Wenn Sie den Namen eines Objekts nicht ändern dürfen, ist das Eingabefeld im Dialogfeld grau unterlegt, der aktuelle Name wird angezeigt und Texteingaben sind nicht möglich.

Hinweis

Wenn Sie während des Editierens den Mauszeiger aus dem Namensfeld herausbewegen und eine andere Aktion durchführen (z.B. einen Menübefehl wählen), wird der Editiervorgang beendet. Der geänderte Namen wird, wenn er zulässig ist, übernommen.

Eine Schritt-für-Schritt-Anleitung zum Umbenennen finden Sie in Umbenennen von Objekten.

Verschieben von Objekten

Mit dem SIMATIC Manager können Sie Objekte von einem Ordner zu einem anderen verschieben, auch wenn dieser in einem anderen Projekt liegt. Beim Verschieben eines Ordners wird dessen gesamter Inhalt mitverschoben.

Hinweis

Die folgenden Objekte können Sie nicht verschieben:

- Verbindungen
- Systemdatenbausteine (SDB) in der Online-Ansicht
- Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB) in der Online-Ansicht

Eine Schritt-für-Schritt-Anleitung zum Verschieben finden Sie in Verschieben von Objekten.

Sortieren von Objekten

Sie können in der Detail-Ansicht (Menübefehl **Ansicht > Details**) die Objekte nach ihren Attributen sortieren lassen. Klicken Sie dazu auf die entsprechende Überschrift des gewünschten Attributes. Bei erneutem Klicken kehrt sich die Sortierreihenfolge um. Bausteine werden entsprechend ihrer numerischen Ergänzung sortiert, z.B. FB 1, FB 2, FB 11, FB 12, FB 21, FC 1.

Voreingestellte Sortier-Reihenfolge (Default-Sortierung):

Wenn Sie ein Projekt neu öffnen, werden die Objekte in der Ansicht "Details" gemäß einer voreingestellten Sortier-Reihenfolge angezeigt. Beispiele:

- Bausteine werden in der Reihenfolge "Systemdaten, OB, FB, FC, DB, UDT, VAT, SFB, SFC" dargestellt.
- Bei Projekt werden erst alle Stationen und dann die S7-Programme angezeigt.

Die Voreinstellung ist also keine alphanumerisch aufsteigende oder absteigende Sortierung für ein Sortierkriterium der Detail-Ansicht.

Wiederherstellen der Default-Sortierung:

Nach einem Umsortieren z. B. durch Klicken auf die Spaltenüberschrift "Objektname", können Sie die voreingestellte Reihenfolge wieder herstellen, wenn Sie wie folgt vorgehen:

- in der Detailansicht auf die Spaltenüberschrift "Typ" klicken.
- das Projekt schließen und erneut öffnen.

Löschen von Objekten

Sie können sowohl Ordner als auch Objekte löschen. Wenn Sie einen Ordner löschen, werden auch alle darin enthaltenen Objekte mitgelöscht.

Der Löschvorgang kann nicht rückgängig gemacht werden. Wenn Sie nicht sicher sind, ob Sie ein Objekt wirklich nicht mehr brauchen, ist es besser, das gesamte Projekt vorher zu archivieren.

Hinweis

Die folgenden Objekte können Sie nicht löschen:

- Verbindungen
- Systemdatenbausteine (SDB) in der Online-Ansicht
- Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB) in der Online-Ansicht

Eine Schritt-für-Schritt-Anleitung zum Löschen finden Sie in Löschen von Objekten.

5.3.5 Auswahl von Objekten im Dialog

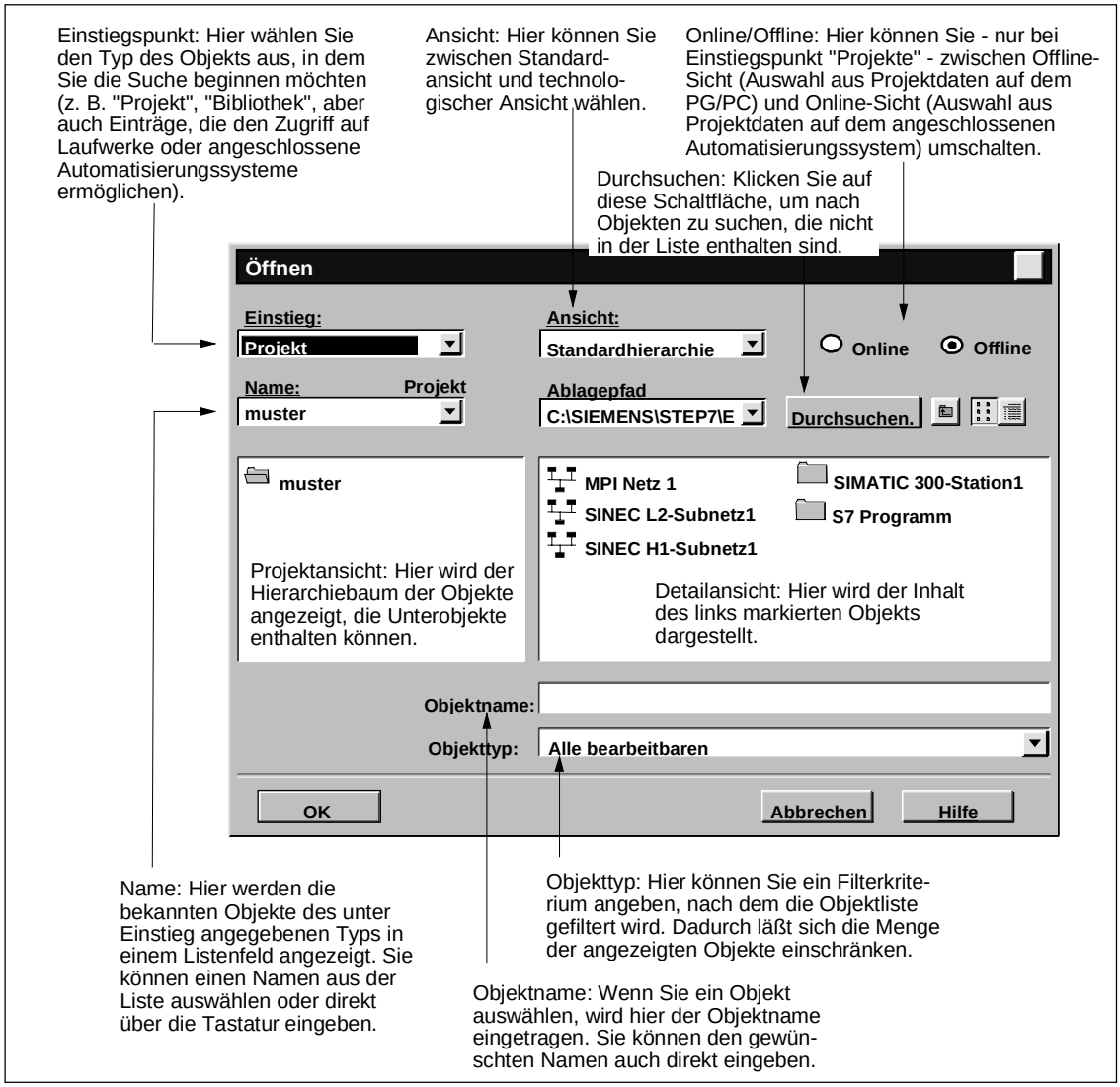
Die Auswahl von Objekten in einem Dialogfeld ist eine bei unterschiedlichen Arbeitsschritten auszuführende Aktion.

Aufruf des Dialogfeldes

Das Dialogfeld wird z. B. in der Hardware-Konfigurierung durch die Menübefehle wie **Station > Neu.../Öffnen...** aufgerufen (Eine Ausnahme bildet das Einstiegsfenster ”SIMATIC Manager”).

Aufbau des Dialogfeldes

Im Dialogfeld haben Sie die im nachfolgenden Bild gezeigten Auswahlmöglichkeiten.


Einstiegspunkt: Hier wählen Sie den Typ des Objekts aus, in dem Sie die Suche beginnen möchten (z. B. "Projekt", "Bibliothek", aber auch Einträge, die den Zugriff auf Laufwerke oder angeschlossene Automatisierungssysteme ermöglichen).
Ansicht: Hier können Sie zwischen Standard-ansicht und technolo-gischer Ansicht wählen.
Online/Offline: Hier können Sie - nur bei Einstiegspunkt "Projekte" - zwischen Offline-Sicht (Auswahl aus Projektdaten auf dem PG/PC) und Online-Sicht (Auswahl aus Projektdaten auf dem angeschlossenen Automatisierungssystem) umschalten.
Durchsuchen: Klicken Sie auf diese Schaltfläche, um nach Objekten zu suchen, die nicht in der Liste enthalten sind.
Öffnen
Einstieg:
Projekt
Ansicht:
Standardhierarchie
Online
Offline
Name:
Projekt
muster
Ablagepfad
C:\SIEMENS\STEP7\E
Durchsuchen.
muster
MPI Netz 1
SINEC L2-Subnetz1
SINEC H1-Subnetz1
SIMATIC 300-Station1
S7 Programm
Projektansicht: Hier wird der Hierarchiebaum der Objekte angezeigt, die Unterobjekte enthalten können.
Detailansicht: Hier wird der Inhalt des links markierten Objekts dargestellt.
Objektname:
Objekttyp:
Alle bearbeitbaren
OK
Abbrechen
Hilfe
Name: Hier werden die bekannten Objekte des unter Einstieg angegebenen Typs in einem Listenfeld angezeigt. Sie können einen Namen aus der Liste auswählen oder direkt über die Tastatur eingeben.
Objekttyp: Hier können Sie ein Filterkrite-rium angeben, nach dem die Objektliste gefiltert wird. Dadurch läßt sich die Menge der angezeigten Objekte einschränken.
Objektname: Wenn Sie ein Objekt auswählen, wird hier der Objektname eingetragen. Sie können den gewün-schten Namen auch direkt eingeben.

5.3.6 Sitzungsgedächtnis

Der SIMATIC Manager kann sich den Fensterinhalt, also die geöffneten Projekte und Bibliotheken, sowie die Fensteranordnung merken.

- Über Menübefehl **Extras > Einstellungen** legen Sie fest, ob Fensterinhalt und Fensteranordnung bei Sitzungsende gespeichert werden. Beim nächsten Sitzungsbeginn werden dieser Fensterinhalt und diese Fensteranordnung wiederhergestellt. In den geöffneten Projekten wird auf den letzten selektierten Behälter positioniert.
- Über Menübefehl **Fenster > Anordnung speichern** werden der aktuelle Fensterinhalt und die aktuelle Fensteranordnung gespeichert.
- Über Menübefehl **Fenster > Anordnung wiederherstellen** werden der mit Menübefehl **Fenster > Anordnung speichern** gespeicherte Fensterinhalt und die Fensteranordnung wiederhergestellt. In den geöffneten Projekten wird auf den letzten selektierten Behälter positioniert.

Hinweis

Der Fensterinhalt von Online-Projekten, der Inhalt des Fensters "Erreichbare Teilnehmer" und der Inhalt des Fensters "S7-Memory Card" werden nicht gespeichert.

Eventuell eingegebene Paßwörter für den Zugriff auf Zielsysteme (S7-300/S7-400) werden über das Sitzungsende hinaus nicht gespeichert.

5.3.7 Ändern der Fensteranordnung von Symboltabellen

Um alle angezeigten Fenster der geöffneten Symboltabellen hintereinander gestaffelt anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Überlappend**.

Um alle angezeigten Fenster der geöffneten Symboltabellen gleichmäßig untereinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Untereinander**.

Um alle angezeigten Fenster der geöffneten Symboltabellen gleichmäßig nebeneinander anzuordnen, wählen Sie den Menübefehl **Fenster > Anordnen > Nebeneinander**.

Um die Symbole der minimierten Fenster gleichmäßig am unteren Rand des Hauptfensters anzuordnen, wählen Sie den Menübefehl **Fenster > Symbole anordnen**.

5.3.8 Speichern und Wiederherstellen der Fensteranordnung

Die STEP 7-Applikationen bieten die Möglichkeit, die aktuelle Anordnung von Fenstern zu speichern und zu einem anderen Zeitpunkt wiederherzustellen. Die Einstellung können Sie über den Menübefehl **Extras > Einstellungen > Allgemein** vornehmen.

Was wird gespeichert?

Wenn Sie die Fensteranordnung speichern, werden folgende Informationen aufgezeichnet:

- Position des Hauptfensters
- Geöffnete Projekte und Bibliotheken und die zugehörigen Fensterpositionen
- Reihenfolge von eventuell übereinanderliegenden Fenstern

Hinweis

Der Fensterinhalt von Online-Projekten, der Inhalt des Fensters “Erreichbare Teilnehmer” und der Inhalt des Fensters “S7-Memory Card” werden nicht gespeichert.

Fensteranordnung speichern

Um die aktuelle Anordnung der Fenster zu speichern, wählen Sie den Menübefehl **Fenster > Anordnung speichern**.

Fensteranordnung wiederherstellen

Um eine gespeicherte Fensteranordnung wieder herzustellen, wählen Sie den Menübefehl **Fenster > Anordnung wiederherstellen**.

Hinweis

Beim Wiederherstellen eines Fensters wird nur der Teil der Hierarchie detailliert dargestellt, in dem das Objekt liegt, das beim Speichern der Fensteranordnung selektiert war.

5.4 Bedienung über Tastureingabe

5.4.1 Bedienung über Tastatureingabe

Internationale Tastenbezeichnung	Deutsche Tastenbezeichnung
HOME-Taste	POS1-Taste
END-Taste	ENDE-Taste
PAGE-UP-Taste	BILD-AUF-Taste
PAGE-DOWN-Taste	BILD-AB-Taste
CTRL-Taste	STRG-Taste
ENTER-Taste	Eingabetaste
DEL-Taste	ENTF-Taste
INSERT-Taste	EINFG-Taste

5.4.2 Tastenkombinationen für Menübefehle

Jeder Menübefehl kann über das Eingeben der entsprechenden ALT-Tastenkombination ausgelöst werden.

Drücken Sie nacheinander folgende Tasten:

- ALT-Taste
- Buchstabe, der im gewünschten Menü unterstrichen ist (z.B. ALT, D für Menü "Datei" - wenn Menü "Datei" in der Menüleiste eingetragen ist). Das Menü wird aufgeschlagen.
- Buchstabe, der im gewünschten Menübefehl unterstrichen ist (z.B. N für Menübefehl "Neu").Handelt es sich um einen Menübefehl mit weiteren Untermenüs, so werden diese daraufhin aufgeschlagen. Verfahren Sie in gleicher Weise, bis Sie den gesamten Menübefehl durch Eingabe der jeweiligen Buchstaben ausgewählt haben.

Mit Eingabe des letzten Buchstabens der Tastenkombination wird der Menübefehl ausgelöst.

Beispiele:

Menübefehle	**Tasten**
Datei > Archivieren	ALT, D, A
Datei > Öffnen	ALT, D, O

Shortcuts für Menübefehle

Funktion		Shortcut
Neu	(Menü "Datei")	CTRL+N
Öffnen	(Menü "Datei")	CTRL+O
Schließen	(Menü "Datei")	-
Übersetzen	(Menü "Datei")	CTRL+B
Drucken	(Objekt) (Menü "Datei")	CTRL+P
Beenden	(Menü "Datei")	ALT+F4
Kopieren	(Menü "Bearbeiten")	CTRL+C
Ausschneiden	(Menü "Bearbeiten")	CTRL+X
Einfügen	(Menü "Bearbeiten")	CTRL+V
Löschen	(Menü "Bearbeiten")	DEL
Alles markieren	(Menü "Bearbeiten")	CTRL+A
Objekteigenschaften	(Menü "Bearbeiten")	ALT+RETURN
Objekt öffnen	(Menü "Bearbeiten")	CTRL+ALT+O
Laden	(Menü "Zielsystem")	CTRL+L
Betriebszustand	(Menü "Zielsystem")	CTRL+I
Aktualisieren	(Menü "Ansicht")	F5
aktualisiert die Statusanzeige der sichtbaren CPUs in der Online-Sicht		CTRL+F5
Einstellungen	(Menü "Extras")	CTRL+ALT+E
Referenzdaten, Anzeigen	(Menü "Extras")	CTRL+ALT+R
Anordnen, Überlappend	(Menü "Fenster")	SHIFT+F5
Anordnen, Horizontal	(Menü "Fenster")	SHIFT+F2
Anordnen, Vertikal	(Menü "Fenster")	SHIFT+F3
Hilfe zum Kontext	(Menü "Hilfe")	F1 (Besteht ein aktueller Kontext, z.B. markierter Menübefehl, wird das entsprechende Hilfethema aufgerufen, andernfalls das Inhaltsverzeichnis der Hilfe.)

5.4.3 Tastenkombinationen für das Bewegen des Cursors

Bewegen des Cursors in der Menüleiste / im PopUp-Menü

Funktion	Tasten
Zur Menüleiste	F10
Zum PopUp-Menü	SHIFT+F10
Zu dem Menü, das den unterstrichenen Buchstaben X enthält	ALT+X
Zugeordneter Menübefehl	Unterstrichener Buchstabe im Menübefehl
Einen Menübefehl nach links	PFEIL LINKS
Einen Menübefehl nach rechts	PFEIL RECHTS
Einen Menübefehl nach oben	PFEIL OBEN
Einen Menübefehl nach unten	PFEIL UNTEN
Aktiviert den markierten Menübefehl	ENTER
Verläßt das Menü bzw. zurück zum Text	ESC

Bewegen des Cursors beim Bearbeiten von Texten

Funktion	Tasten
Eine Zeile nach oben oder ein Zeichen nach links in einem Text, der nur aus einer Zeile besteht	PFEIL OBEN
Eine Zeile nach unten oder ein Zeichen nach rechts in einem Text, der nur aus einer Zeile besteht	PFEIL UNTEN
Ein Zeichen nach rechts	PFEIL RECHTS
Ein Zeichen nach links	PFEIL LINKS
Ein Wort nach rechts	CTRL+PFEIL RECHTS
Ein Wort nach links	CTRL+PFEIL LINKS
Zum Anfang der Zeile	HOME
Zum Ende der Zeile	ENDE
Einen Bildschirm nach oben	PAGE-UP
Einen Bildschirm nach unten	PAGE-DOWN
Zum Textanfang	CTRL+HOME
Zum Textende	CTRL+END

Bewegen des Cursors in Dialogfeldern

Funktion	Tasten
Zum nächsten Eingabefeld (von links nach rechts und von oben nach unten)	TAB
Ein Eingabefeld zurück	SHIFT+TAB
Zu dem Eingabefeld, das den unterstrichenen Buchstaben X enthält	ALT+X
Markiert in Auswahlliste	RICHTUNGSTASTEN
Öffnet eine Auswahlliste	ALT+PFEIL UNTEN
Markiert ein Objekt bzw. macht Markierung rückgängig	LEERTASTE
Bestätigt Einträge und schließt das Dialogfeld (Schaltfläche "OK")	ENTER
Schließt das Dialogfeld, ohne die Auswahl zu speichern (Schaltfläche "Abbrechen")	ESC

5.4.4 Tastenkombinationen für das Markieren von Texten

Funktion	Tasten
Ein Zeichen nach rechts	SHIFT+PFEIL RECHTS
Ein Zeichen nach links	SHIFT+PFEIL LINKS
Bis zum Anfang der Kommentarzeile	SHIFT+HOME
Bis zum Ende der Kommentarzeile	SHIFT+END
Eine Zeile nach oben	SHIFT+PFEIL OBEN
Eine Zeile nach unten	SHIFT+PFEIL UNTEN
Einen Bildschirm nach oben	SHIFT+PAGE-UP
Einen Bildschirm nach unten	SHIFT+PAGE-DOWN
Bis zum Dateianfang	CTRL+SHIFT+HOME
Bis zum Dateiende	CTRL+SHIFT+END

5.4.5 Tastenkombinationen für den Zugriff auf Online-Hilfe

Funktion	Tasten
Öffnet die Hilfe	F1 (Besteht ein aktueller Kontext, z.B. markierter Menübefehl, wird das entsprechende Hilfethema aufgerufen, andernfalls das Inhaltsverzeichnis der Hilfe.)
Aktivieren des Fragezeichen-Symbols für kontextsensitive Hilfe	SHIFT+F1
Schließt das Hilfefenster, zurück zum Toolfenster	ALT+F4

5.4.6 Tastenkombinationen für das Umschalten zwischen Fenstertypen

Funktion	Tasten
Umschalten zwischen Teilfenstern	F6
Zurückschalten auf das vorherige Teilfenster, wenn kein andockbares Fenster vorhanden ist	Shift+F6
Umschalten zwischen Dokumentfenster und andockbarem Fenster des Dokuments (z.B. Variablendeklarationsfenster) Wenn es keine andockbaren Fenster gibt, wird auf das vorherige Teilfenster zurückgeschaltet.	Shift+F6
Umschalten zwischen Dokumentfenstern	Ctrl+F6
Zurückschalten auf das vorherige Dokumentfenster	Shift+Ctrl+F6
Umschalten zwischen Nichtdokumentfenstern (Applikationsrahmen und andockbare Fenster des Applikationsrahmens; bei Rückkehr zum Rahmen wird zu dem Dokumentfenster umgeschaltet, das zuletzt aktiv war)	Alt+F6
Zurückschalten auf das vorherige Nichtdokumentfenster	Shift+Alt+F6
Schließen des aktiven Fensters	Ctrl+F4

6 Einrichten und Bearbeiten des Projekts

6.1 Projektstruktur

Projekte dienen dazu, die bei der Erstellung einer Automatisierungslösung anfallenden Daten und Programme geordnet abzulegen. Die in einem Projekt zusammengefaßten Daten umfassen insbesondere:

- Konfigurationsdaten über den Hardware-Aufbau und Parametrierungsdaten für Baugruppen,
- Projektierungsdaten für die Kommunikation über Netze und
- Programme für programmierbare Baugruppen.

Die Hauptaufgabe bei der Erstellung eines Projekts sind die Bereitstellung dieser Daten und die Programmierung.

Daten werden in einem Projekt in Form von Objekten abgelegt. Die Objekte sind innerhalb eines Projekts in einer Baumstruktur (Projekthierarchie) angeordnet. Die Darstellung der Hierarchie im Projektfenster ähnelt der des Windows 95-Explorers. Lediglich die Objektsymbole haben ein anderes Aussehen.

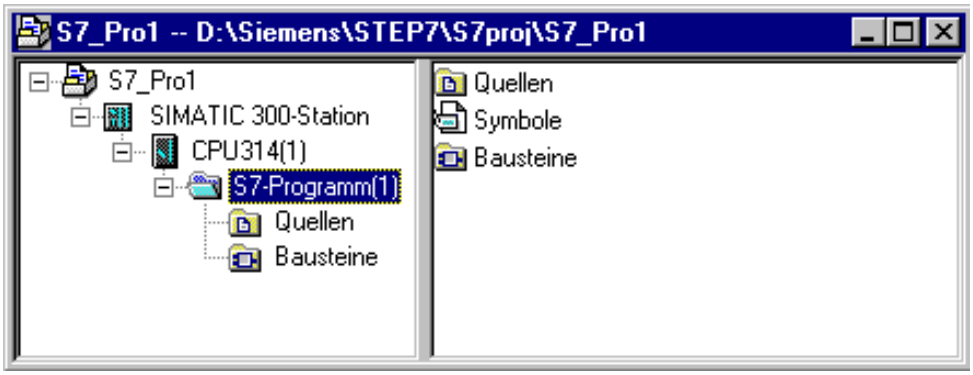
Die Spitze der Projekthierarchie ist wie folgt aufgebaut:

1. Stufe : Projekt
2. Stufe : Subnetze, Stationen oder S7-/M7-Programme
3. Stufe: abhängig vom zugehörigen Objekt der 2. Stufe.

Projektfenster

Das Projektfenster ist zweigeteilt. In der linken Hälfte wird die Baumstruktur des Projekts dargestellt. In der rechten Hälfte wird der Inhalt des links markierten Objekts in der gewählten Ansicht angezeigt (Große Symbole, kleine Symbole, Liste oder Details) .

Klicken Sie im linken Teil des Fensters auf die Kästchen mit einem Pluszeichen, um die vollständige Baumstruktur des Projekts sichtbar zu machen. Sie erhalten dann eine Darstellung ähnlich der im nachfolgenden Bild.



An der Spitze der Objekt-Hierarchie steht hier das Objekt ”S7_Pro1” als Symbol für das gesamte Projekt. Es kann zum Anzeigen der Projekteigenschaften benutzt werden und dient als Ordner für Netze (zum Projektieren von Netzen), Stationen (zum Konfigurieren der Hardware) sowie für S7- oder M7-Programme (zur Software-Erstellung). Die enthaltenen Objekte werden rechts im Projektfenster angezeigt, wenn Sie das Projektsymbol markieren. Die Objekte an der Spitze der Hierarchie (dazu gehören neben Projekten auch Bibliotheken) bilden die Einstiegspunkte in Dialogfeldern zur Auswahl von Objekten.

Projektsicht

Sie können sich die Projektstruktur für den Datenbestand auf dem Erstellsystem in der Ansicht “offline” und für den zugehörigen Datenbestand auf dem Zielsystem in der Ansicht “online” in Projektfenstern anzeigen lassen.

Als weitere einstellbare Sicht steht, wenn das entsprechende Optionspaket geladen ist, die des Anlagenmanagers zur Verfügung.

Hinweis

Das Konfigurieren von Hardware und das Projektieren von Netzen ist nur in der Ansicht “offline” durchführbar.

6.2 Einrichten eines Projekts

6.2.1 Anlegen eines Projekts

Um Ihre Automatisierungsaufgabe im Rahmen einer Projektverwaltung zu lösen, legen Sie zunächst ein neues Projekt an. Das neue Projekt wird in dem Verzeichnis angelegt, das Sie nach Anwahl des Menübefehls **Extras > Einstellungen** im Register “Allgemein” für die Projekte eingestellt haben.

Hinweis

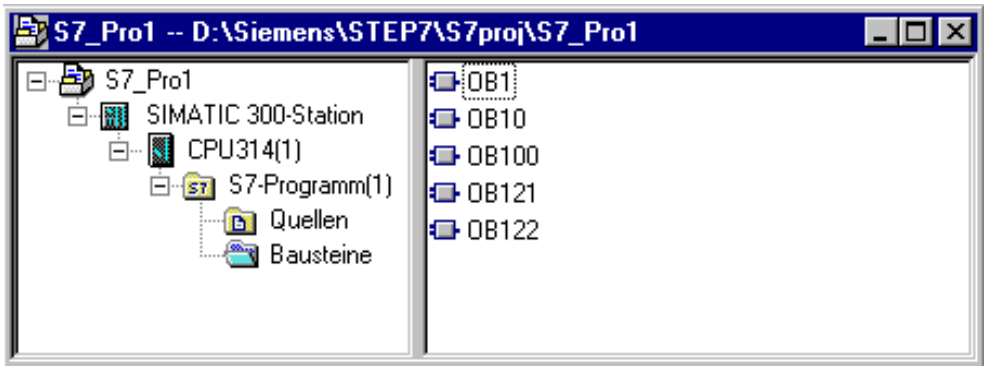
Der SIMATIC Manager läßt Namen zu, die länger als 8 Zeichen sind. Der Name des Projektverzeichnisses wird jedoch auf 8 Zeichen abgeschnitten. Projektnamen müssen sich daher in den ersten 8 Zeichen unterscheiden. Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Eine Schritt-für-Schritt-Anleitung zum Anlegen eines Projekts finden Sie in Anlegen eines Projekts manuell oder in Anlegen eines Projekts über Assistent

Anlegen eines Projekts über Assistenten

Ein neues Projekt können Sie am einfachsten mit dem Assistenten 'Neues Projekt' anlegen. Zum Aufruf wählen Sie den Menübefehl **Datei > Assistent 'Neues Projekt'**. Der Assistent fragt die benötigten Angaben mit Hilfe von Dialogfeldern ab und legt dann das Projekt für Sie an. Neben Station, CPU, Programm-, Quellen-, Bausteinordner und OB1 können Sie dort bereits OBs für die Fehler- und Alarmbearbeitung auswählen.

Nachfolgendes Bild zeigt Beispiel für ein mit dem Assistenten angelegtes Projekt.



Manuelles Anlegen eines Projekts

Es besteht aber auch die Möglichkeit, ein neues Projekt über den Menübefehl **Datei > Neu** im SIMATIC Manager anzulegen. Es enthält bereits das Objekt “MPI-Subnetze”.

Alternativen für die Weiterarbeit

Bei der Wahl der weiteren Bearbeitungsreihenfolge eines Projekts haben Sie bestmögliche Flexibilität. Nach dem Anlegen eines Projekts können Sie beispielsweise

- zunächst die Hardware konfigurieren und anschließend die Software dafür erstellen oder aber
- unabhängig von einer konfigurierten Hardware mit der Software-Erstellung beginnen.

Alternative 1: Zunächst Hardware konfigurieren

Wenn Sie zunächst die Hardware konfigurieren möchten, gehen Sie wie in Band 2 des Handbuchs "Hardware-Konfiguration mit STEP 7" beschrieben vor. Nach dem Konfigurieren sind die für die Software-Erstellung erforderlichen Ordner ”S7-Programm” bzw. ”M7-Programm” bereits eingefügt. Dann fahren Sie mit dem Einfügen der zur Programmerstellung erforderlichen Objekte fort. Erstellen Sie anschließend die Software für die programmierbaren Baugruppen.

Alternative 2: Zunächst Software erstellen

Sie können auch ohne vorherige Hardware-Konfigurierung Software erstellen und die Konfigurierung zu einem späteren Zeitpunkt nachholen. Der Hardware-Aufbau einer Station braucht zur Eingabe von Programmen noch nicht festgelegt sein.

Die prinzipielle Vorgehensweise ist wie folgt:

1. Fügen Sie die erforderlichen Software-Ordner (S7 / M7-Programm ohne Station und CPU) in Ihr Projekt ein.
   Dabei entscheiden Sie lediglich, ob der Ordner "Programme" für S7-Hardware oder M7-Hardware enthalten soll.
2. Erstellen Sie anschließend die Software für die programmierbaren Baugruppen.
3. Konfigurieren Sie Ihre Hardware.
4. Ordnen Sie das M7- bzw. S7-Programm einer CPU zu, nachdem Sie die Hardware konfiguriert haben.

6.2.2 Einfügen von Stationen

In einem Projekt repräsentiert die Station den Hardware-Aufbau des Automatisierungssystems und beinhaltet die Daten zur Konfigurierung und Parametrierung der einzelnen Baugruppen.

Neue Projekte, die mit dem Assistenten ’Neues Projekt’ angelegt wurden, enthalten bereits eine Station. Andernfalls können Sie die Station über den Menübefehl **Einfügen > Station** erzeugen.

Folgende Stationen stehen zur Auswahl:

- SIMATIC 300-Station
- SIMATIC 400-Station
- SIMATIC H-Station
- SIMATIC PC-Station
- PC/PG
- SIMATIC S5
- Andere Stationen, d. h. nicht SIMATIC S7/M7, SIMATIC S5

Die Station wird mit einer voreingestellten Bezeichnung (z. B. SIMATIC 300-Station (1), SIMATIC 300-Station (2) usw.) eingefügt. Sie können die Bezeichnung von Stationen durch aussagekräftige Namen ersetzen.

Eine Schritt-für-Schritt-Anleitung zum Einfügen finden Sie in Einfügen einer Station.

Hardware-Konfigurierung durchführen

Über die Hardware-Konfigurierung legen Sie mit Hilfe eines Baugruppenkatalogs die CPU und alle in Ihrer Steuerung enthaltenen Baugruppen fest. Die Hardware-Konfigurierung starten Sie durch Doppelklick auf die Station.

Nach dem Speichern und Beenden der Hardware-Konfigurierung wird zu jeder programmierbaren Baugruppe, die Sie bei der Konfigurierung angelegt haben, automatisch ein S7- bzw. M7-Programm als Software-Ordner sowie eine Verbindungstabelle (Objekt ”Verbindungen”) angelegt. Projekte, die mit dem Assistenten ’Neues Projekt’ angelegt wurden, enthalten diese Objekte bereits von Anfang an.

Eine Schritt-für-Schritt-Anleitung zum Konfigurieren finden Sie in Konfigurieren der Hardware, ausführliche Informationen in Grundsätzliche Schritte beim Konfigurieren einer Station.

Verbindungstabelle erstellen

Eine (leere) Verbindungstabelle (Objekt ”Verbindungen”) wird automatisch zu jeder programmierbaren Baugruppe angelegt. Die Verbindungstabelle wird zur Definition von Kommunikationsverbindungen zwischen programmierbaren Baugruppen in einem Netz verwendet. Nach dem Öffnen wird ein Fenster aufgeblendet und darin eine Tabelle zur Definition von Verbindungen zwischen programmierbaren Baugruppen.

Ausführliche Informationen finden Sie in Vernetzen von Stationen innerhalb eines Projekts.

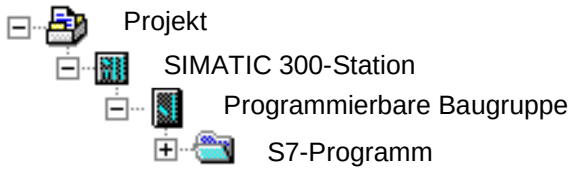
Nächste Schritte

Nach dem Erstellen der Hardware-Konfiguration können Sie die Software für Ihre programmierbaren Baugruppen erstellen (siehe auch Einfügen eines S7-/M7-Programms).

6.2.3 Einfügen eines S7- / M7-Programms

Die Software für programmierbare Baugruppen wird in Objektordnern abgelegt. Für SIMATIC S7-Baugruppen heißt ein solcher Objektordner ”S7-Programm”, für SIMATIC M7-Baugruppen ”M7-Programm”.

Nachfolgendes Bild zeigt als Beispiel ein S7-Programm in einer programmierbaren Baugruppe einer SIMATIC 300-Station.



Bereits angelegte Komponenten

Zu jeder programmierbaren Baugruppe wird automatisch ein S7- / M7-Programm als Ordner für die Software angelegt.

Das S7-Programm enthält bereits:

- Symboltabelle (Objekt ”Symbole”),
- einen Ordner ”Bausteine” für Bausteine mit erstem Baustein,
- einen Ordner ”Quellen” für Quellprogramme.

Das M7-Programm enthält bereits:

- Symboltabelle (Objekt ”Symbole”),
- einen Ordner ”Bausteine”.

Erstellen von S7-Bausteine

Sie wollen AWL-, KOP- oder FUP-Programme erstellen. Dazu markieren Sie das bereits angelegte Objekt ”Bausteine” und klicken dann auf den Menübefehl **Einfügen > S7-Baustein**. In einem Folgemenü können Sie den Bausteintyp auswählen (Datenbaustein, Datentyp (UDT), Funktion, Funktionsbaustein, Organisationsbaustein, Variablentabelle (VAT)).

Nach dem Öffnen des (leeren) Bausteins können Sie das AWL-, KOP- oder FUP-Programm eingeben. Weitere Information dazu finden Sie in Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen und in den Handbüchern zu AWL, KOP und FUP.

Hinweis

Das Objekt Systemdaten (SDB), das Sie eventuell in Anwenderprogrammen vorfinden, wird vom System erzeugt. Sie können das Objekt zwar öffnen, den Inhalt aus Konsistenzgründen aber nicht ändern. Es dient dazu, nach dem Laden eines Programms Konfigurationsänderungen vornehmen und diese auf das Zielsystem nachladen zu können.

Verwenden von Bausteinen aus Standardbibliotheken

Für die Erstellung von Anwenderprogrammen können Sie auch Bausteine aus den Standardbibliotheken benutzen, die im Lieferumfang enthalten sind. Auf Bibliotheken greifen Sie über den Menübefehl **Datei > Öffnen** zu. Weitere Hinweise zum Benutzen von Standardbibliotheken sowie zum Anlegen eigener Bibliotheken finden Sie in Arbeiten mit Bibliotheken in der Online-Hilfe.

Erstellen von Quellen/CFC-Pläne

Sie wollen eine Quelle in einer bestimmten Programmiersprache oder einen CFC-Plan erstellen. Dazu markieren Sie im S7-Programm das Objekt ”Quellen” bzw. ”Pläne” und klicken anschließend auf den Menübefehl **Einfügen > S7-Software**. Wählen Sie im Folgemenü die zur Programmiersprache passende Quelle aus. Nach dem Öffnen der leeren Quelle können Sie das Programm eingeben. Weitere Hinweise finden Sie in Grundlagen zum Programmieren in AWL-Quellen.

Erstellen von Programmen für M7

Sie wollen Programme für das Betriebssystem RMOS einer programmierbaren Baugruppe aus dem M7-Spektrum erstellen. Dazu markieren Sie das M7-Programm und klicken anschließend auf den Menübefehl **Einfügen > M7-Software**. Wählen Sie im Folgemenü das zur Programmiersprache bzw. zum Betriebssystem passende Objekt aus. Nach dem Öffnen des angelegten Objekts gelangen Sie in die zugehörige Erstellumgebung.

Erstellen einer Symboltabelle

Eine (leere) Symboltabelle (Objekt ”Symbole”) wird beim Erzeugen des S7- / M7-Programms automatisch angelegt. Nach dem Öffnen wird das Fenster ”Symbol Editor” aufgeblendet und darin die Symboltabelle angezeigt. Weitere Hinweise finden Sie in Eingeben mehrerer globaler Symbole in der Symboltabelle.

Einfügen externer Quellen

Sie können Quelldateien mit beliebigen ASCII-Editoren erstellen und bearbeiten. Diese Dateien können Sie anschließend in ein Projekt importieren und in einzelne Bausteine übersetzen.

Die bei der Übersetzung einer importierten Quelle entstehenden Bausteine werden im Ordner ”Bausteine” abgelegt.

Weitere Hinweise finden Sie in Einfügen von externen Quellen.

6.3 Bearbeiten eines Projekts

6.3.1 Bearbeiten eines Projekts

Öffnen eines Projekts

Um ein Projekt zu öffnen, geben Sie zunächst den Menübefehl **Datei > Öffnen** ein. Wählen Sie dann ein Projekt in den Folgedialogen aus. Danach wird das Projektfenster geöffnet.

Hinweis

Wenn Sie das Projekt nicht in der Projektliste angezeigt bekommen, betätigen Sie die Schaltfläche “Durchsuchen”. In dem gleichnamigen Dialogfeld können Sie nach weiteren Projekten suchen und gefundene Projekte in die Projektliste aufnehmen. Die Einträge in der Projektliste können Sie mit dem Menübefehl **Datei > Verwalten** ändern.

Kopieren eines Projekts

Ein Projekt kopieren Sie, indem Sie dieses über den Menübefehl **Datei > Speichern unter** unter einen anderen Namen abspeichern.

Projektteile wie Stationen, Programme, Bausteine usw. kopieren Sie mit dem Menübefehl **Bearbeiten > Kopieren**.

Eine Schritt-für-Schritt-Anleitung zum Kopieren eines Projekts finden Sie in Kopieren eines Projekts und Kopieren eines Projektteils.

Löschen eines Projekts

Ein Projekt löschen Sie über den Menübefehl **Datei > Löschen**.

Projektteile wie Stationen, Programme, Bausteine usw. löschen Sie mit dem Menübefehl **Bearbeiten > Löschen**.

Eine Schritt-für-Schritt-Anleitung zum Löschen eines Projekts finden Sie in Löschen eines Projekts und Löschen eines Projektteils.

6.3.2 Texte mehrsprachig verwalten

STEP 7 bietet die Möglichkeit, Texte, die in einem Projekt einsprachig abgelegt sind, zu exportieren, übersetzen zu lassen, wieder zu importieren und in der übersetzten Sprache anzeigen zu lassen.

Die folgenden Arten von Texten können mehrsprachig verwaltet werden.

- Kommentare und Titel
  - Bausteintitel und Bausteinkommentare
  - Netzwerktitel und Netzwerkkommentare
  - Zeilenkommentare aus AWL-Programmen
  - Kommentare aus Symboltabellen, Variablendeklarationstabellen, anwenderdefinierten Datentypen und Datenbausteinen
  - Kommentare, Zustandsnamen und Transitionsnamen in HiGraph-Programmen
  - Erweiterungen zu Schrittnamen und Schrittkommentare in S7-GRAPH-Programmen
- Anzeigetexte
  - Meldungstexte, die von STEP 7, S7-GRAPH, S7-HiGraph oder S7-PDIAG generiert werden.
  - System-Textbibliotheken.

Export

Der Export wird für alle Bausteine und Symboltabellen ausgeführt, die unter dem angewählten Objekt liegen. Für jeden Texttyp wird eine Exportdatei erzeugt. Diese enthält je eine Spalte für die Quell- und für die Zielsprache. Die Texte in der Quellsprache dürfen nicht geändert werden.

Import

Beim Import wird der Inhalt der Spalten für die Zielsprache (rechte Spalte) in das angewählte Projekt übernommen. Es werden dabei nur die Texte übernommen, für die in der Spalte für die Quellsprache ein Übereinstimmung mit einem vorhandenen Text festgestellt wird.

Sprachwechsel

Beim Sprachwechsel können alle Sprachen ausgewählt werden, die beim Import in das angewählte Projekt angegeben wurden. Der Sprachwechsel wird für die angewählten Objekte ausgeführt.

Sprache löschen

Beim Löschen einer Sprache werden alle Texte in dieser Sprache aus der internen Datenhaltung gelöscht.

Es sollte immer eine Sprache als Bezugssprache in Ihrem Projekt vorhanden sein. Das kann beispielsweise Ihre Landessprache sein. Diese Sprache sollten Sie nicht löschen. Beim Exportieren und Importieren stellen Sie diese Bezugssprache immer als Quellsprache ein. Die Zielsprache stellen Sie nach Wunsch ein.

Prinzipielle Vorgehensweise

Erzeugen Sie die Export-Dateien für die Übersetzung (Menübefehl **Extras > Texte mehrsprachig verwalten > Exportieren**).
Übersetzen Sie die Texte.
Importieren Sie die übersetzten Dateien (Menübefehl **Extras > Texte mehrsprachig verwalten > Importieren**).
Wählen Sie die Sprache, in der die Texte angezeigt werden sollen (Menübefehl **Extras > Texte mehrsprachig verwalten > Sprachwechsel**).

7 Festlegen von Symbolen

7.1 Absolute und symbolische Adressierung

In einem STEP 7-Programm arbeiten Sie mit Operanden wie E/A-Signalen, Merkern, Zählern, Zeiten, Datenbausteinen und Funktionsbausteinen. Sie können diese Operanden in Ihrem Programm absolut adressieren, die Lesbarkeit von Programmen steigt jedoch erheblich, wenn Sie dafür Symbole (z. B. Motor_A_Ein oder Bezeichnungen gemäß eines in Ihrer Branche üblichen Kennzeichnungssystems) benutzen. In Ihrem Anwenderprogramm läßt sich dann ein Operand über dieses Symbol ansprechen.

Absolute Adresse

Eine absolute Adresse besteht aus einem Operandenkennzeichen und einer Adresse (z. B. A 4.0, E 1.1, M 2.0, FB 21).

Symbolische Adressierung

Sie können Ihr Programm übersichtlicher gestalten und die Fehlerbehebung vereinfachen, wenn Sie die absoluten Adressen mit symbolischen Namen versehen.

STEP 7 kann automatisch die symbolischen Namen in die erforderlichen absoluten Adressen übersetzen. Wenn Sie auf ARRAYs, STRUCTs, Datenbausteine, Lokaldaten, Codebausteine und anwenderdefinierte Datentypen vorzugsweise über symbolische Namen zugreifen, dann müssen Sie den absoluten Adressen jedoch zunächst symbolische Namen zuordnen, bevor Sie die Daten symbolisch adressieren können.

Sie können beispielsweise dem Operanden A 4.0 den symbolischen Namen MOTOR_EIN zuordnen und dann MOTOR_EIN als Adresse in einer Programmanweisung verwenden. Über symbolische Adressen können Sie leichter erkennen, inwieweit Elemente des Programms mit den Komponenten Ihres Prozeßsteuerungsprojekts übereinstimmen.

Hinweis

In einem symbolischen Namen (Variablenbezeichnung) sind zwei Unterstriche nacheinander (wie z. B.: MOTOR__EIN) nicht erlaubt.

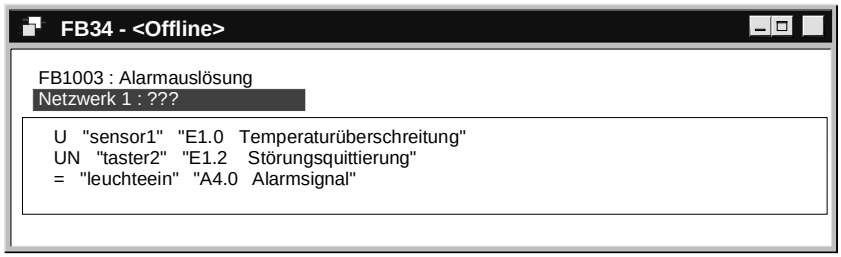
Unterstützung bei der Programmeingabe

In den Programmiersprachen KOP, FUP und AWL können Sie Adressen, Parameter und Bausteinnamen absolut oder als Symbol angeben.

Mit dem Menübefehl **Ansicht > Symbolische Darstellung** schalten Sie zwischen der Darstellung absoluter und symbolischer Adressierung um.

Zur leichteren Programmierung mit symbolischer Adressierung können Sie sich zu den vorkommenden Symbolen die zugehörige absolute Adresse mit dem Symbolkommentar anzeigen lassen. Dieses erreichen Sie über den Menübefehl **Ansicht > Symbolinformation**. Dadurch wird hinter jeder AWL-Anweisung der Zeilenkommentar entsprechend ersetzt. Die Anzeige können Sie nicht bearbeiten; Änderungen müssen Sie in der Symboltabelle bzw. in der Variablendeklarationstabelle vornehmen.

Nachfolgendes Bild zeigt Ihnen eine Sybolinformation in AWL.



Beim Ausdruck des Bausteins wird die aktuelle Bildschirmdarstellung mit Anweisungskommentar oder Symbolkommentar wiedergegeben.

7.2 Globale und lokale Symbole

Ein Symbol ermöglicht Ihnen, anstelle von absoluten Adressen mit einer aussagekräftigen Bezeichnung zu arbeiten. Mit der Kombination aus kurzen Symbolen und ausführlicheren Kommentaren können Sie sowohl die Anforderung einer effektiven Programmerstellung als auch einer guten Programmdokumentation erfüllen.

Man unterscheidet zwischen lokalen und globalen Symbolen:

	Globale Symbole	**Lokale Symbole**
Gültigkeitsbereich	• gelten im gesamten Anwenderprogramm • können von allen Bausteinen benutzt werden, • Haben in allen Bausteinen die gleiche Bedeutung, • Die Bezeichnung muß im gesamten Anwenderprogramm eindeutig sein.	• sind nur in dem Baustein bekannt, in dem sie definiert wurden. • Die gleiche Bezeichnung kann in verschiedenen Bausteinen für unterschiedliche Zwecke verwendet werden.
Zulässige Zeichen	• Buchstaben, Ziffern, Sonderzeichen, • Umlaute außer 0x00, 0xFF und Anführungszeichen; • Symbol muß bei Verwendung von Sonderzeichen in Anführungs- und Schlußzeichen gesetzt werden.	• Buchstaben, • Ziffern, • Unterstrich (_),
Einsatz	Sie können globale Symbole definieren für: • E/A-Signale (E, EB, EW, ED, A, AB, AW, AD) • Peripherie-Eingänge / -Ausgänge (PE, PA) • Merker (M, MB, MW, MD) • Zeiten (T)/ Zähler (Z) • Codebausteine (OB, FB, FC, SFB, SFC) • Datenbausteine (DB) • Anwenderdefinierte Datentypen • Variablentabelle	Lokale Symbole können Sie definieren für: • Bausteinparameter (Eingangs-, Ausgangs- und Durchgangsparameter), • statische Daten eines Bausteins • temporäre Daten eines Bausteins.
Ort der Definition	Symboltabelle.	Variablendeklarationstabelle des Bausteins.

7.3 Darstellung von globalen oder lokalen Symbolen

Sie können die globalen und lokalen Symbole im Anweisungsteil eines Programms folgendermaßen auseinanderhalten:

- Symbole aus der Symboltabelle (global) werden in Anführungszeichen **"..."** dargestellt.
- Symbole aus der Variablendeklarationstabelle des Bausteins (bausteinlokal) werden mit vorangestelltem "#" dargestellt.

Die Kennzeichnung in Anführungszeichen oder mit "#" müssen Sie nicht eingeben. Bei der Programmeingabe in KOP, FUP oder AWL wird die Kennzeichnung nach der Syntaxprüfung entsprechend ergänzt.

Wenn jedoch Verwechslungen denkbar sind, weil z.B. gleiche Symbole in Symboltabelle und Variablendeklaration verwendet wurden, müssen Sie, wenn Sie das globale Symbol verwenden wollen, dieses explizit kennzeichnen. Nicht gekennzeichnete Symbole werden in diesem Fall als bausteinlokale Variablen interpretiert.

Die Kennzeichnung globaler Symbole ist zudem erforderlich, wenn das Symbol Leerzeichen enthält.

Bei der Programmierung in einer AWL-Quelle gelten die gleichen Sonderzeichen und Regeln ihrer Verwendung. Die Kennzeichnungen bei der quellorientierten Eingabe werden hier nicht automatisch ergänzt, sind jedoch nur bei Verwechslungsgefahr erforderlich.

Hinweis

Mit dem Menübefehl **Ansicht > Symbolische Darstellung** schalten Sie zwischen der Anzeige der vereinbarten globalen Symbole und der zugehörigen absoluten Adresse um.

7.4 Einstellung des Operandenvorrangs

Im Dialogfeld zu den Eigenschaften des S7-Programms können Sie einstellen, ob nach Änderungen in der Symboltabelle beim Öffnen der Bausteine das Symbol oder der Absolutwert maßgeblich sein soll. In STEP 7-Versionen älter als V5 ist stets der Absolutwert maßgeblich. Beim Bausteinaufruf CALL ist immer nur der Absolutwert maßgeblich.

Beispiel

Ein gespeicherter Baustein enthält die Anweisung "U Symbol_A", wobei in der Symboltabelle Symbol_A für den Absolutwert E0.1 definiert ist. Nun wird die Symboltabelle geändert und anschließend der Baustein erneut geöffnet. Die Einstellung des Operandenvorrangs wirkt sich dann wie folgt auf diese Anweisung aus:

Operanden-vorrang	**Änderung der Zuordnung "Symbol_A = E0.1"**	**Anweisung nach Öffnen des Bausteins**	**Erläuterung**
Absolutwert	Symbol_A = E0.2	U E0.1	In der Anweisung wird der Absolutwert E0.1 angezeigt, da ihm kein Symbol mehr zugeordnet ist.
Absolutwert	Symbol_B = E0.1	U Symbol_B	In der Anweisung wird das neue Symbol für den weiterhin gültigen Absolutwert E0.1 angezeigt.
Symbol	Symbol_A = E0.2	U Symbol_A	Die Anweisung bleibt bestehen. Eine Meldung zur geänderten Symbolzuordnung wird angezeigt.
Symbol	Symbol_B = E0.1	U Symbol_A	Die Anweisung wird als fehlerhaft gekennzeichnet (rote Schrift), da Symbol_A nicht mehr definiert ist.

7.5 Symboltabelle für globale Symbole

7.5.1 Symboltabelle für globale Symbole

In der Symboltabelle definieren Sie globale Symbole.

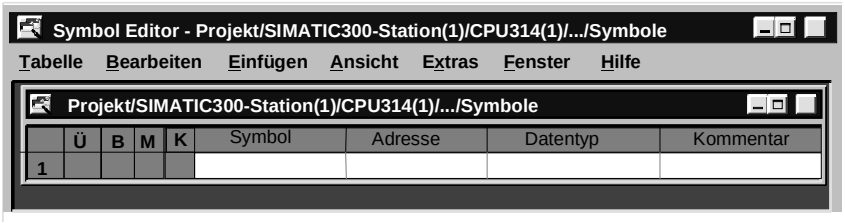
Eine leere Symboltabelle (Objekt ”Symbole”) wird automatisch erzeugt, wenn Sie ein S7- oder M7-Programm anlegen.

Gültigkeitsbereich

Die Symboltabelle gilt genau für die Baugruppe, der Sie das Programm zuordnen. Wenn Sie in verschiedenen CPUs die gleichen Symbole verwenden wollen, müssen Sie selbst dafür sorgen, daß die entsprechenden Einträge in den zugehörigen Symboltabellen übereinstimmen (z. B. durch Kopieren).

7.5.2 Struktur und Bestandteile der Symboltabelle

Struktur der Symboltabelle


Symbol Editor - Projekt/SIMATIC300-Station(1)/CPU314(1)/.../Symbole
Tabelle Bearbeiten Einfügen Ansicht Extras Fenster Hilfe
Projekt/SIMATIC300-Station(1)/CPU314(1)/.../Symbole
Ü B M K Symbol Adresse Datentyp Kommentar
1

Spalten Ü/B/M/K

Die Spalten Ü/B/M/K zeigen an, ob einem Symbol spezielle Objekteigenschaften zugewiesen wurden:

- Ü bedeutet, für das Symbol wurden Fehlerdefinitionen zur Prozeßdiagnose mit dem Optionspaket S7-PDIAG (V5) angelegt.
- B bedeutet, das Symbol kann mit WinCC bedient und beobachtet werden.
- M bedeutet, dem Symbol wurde eine symbolbezogene Meldung (SCAN) zugeordnet.
- K bedeutet, dem Symbol wurden Kommunikationseigenschaften zugeordnet (nur mit NCM anwählbar).

Symbol

Der Name des Symbols darf maximal 24 Zeichen lang sein. Eine Symboltabelle kann maximal 16380 Symbole enthalten.

Für Operanden aus Datenbausteinen (DBD, DBW, DBB, DBX) können Sie in der Symboltabelle keine Namen vergeben. Deren Namen sind durch die Deklaration in den Datenbausteinen festgelegt.

Für Organisationsbausteine (OBs) und einige Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs) existieren vordefinierte Symboltabelleneinträge, die Sie beim Bearbeiten der Symboltabelle Ihres S7-Programms in diese importieren können. Die Importdatei finden Sie im STEP 7-Verzeichnis unter ...\S7data\Symbol\Symbol.sdf.

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden.

Beispiel: Eingang E 12.1

Die Syntax der Adresse wird bei der Eingabe überprüft. Außerdem wird geprüft, ob die Adresse dem angegebenen Datentyp zugeordnet werden darf.

Datentyp

Sie können unter den verschiedenen Datentypen wählen, die Ihnen STEP 7 zur Verfügung stellt. Das Feld wird mit einem Default-Datentyp vorbesetzt, den Sie jedoch ändern können. Falls Ihre Änderung nicht zur Adresse paßt oder die Syntax falsch ist, erscheint beim Verlassen des Feldes eine Fehlermeldung.

Kommentar

Allen Symbolen können Sie Kommentare zuordnen. Mit einer Kombination aus kurzen Symbolen und ausführlichen Kommentaren können Sie sowohl Anforderungen an eine gute Programmdokumentation als auch an eine effektive Programmerstellung erfüllen. Ein Kommentar kann maximal 80 Zeichen lang sein.

Umsetzung in C-Variable

Aus der Symboltabelle eines M7-Programms können Sie Symbole auswählen und in Verbindung mit der Optionssoftware ProC/C++ in korrespondierende C-Variablen umsetzen.

7.5.3 Zulässige Adressen und Datentypen in der Symboltabelle

Es ist nur eine Notation über die gesamte Symboltabelle möglich. Die Umschaltung zwischen Deutsch (vormals SIMATIC) und Englisch (vormals IEC) muß im SIMATIC Manager erfolgen, unter **Extras > Einstellungen** im Register "Sprache".

Englisch	Deutsch	Erläuterung:	Datentyp:	Adreßbereich:
I	E	Eingangsbit	BOOL	0.0..65535.7
IB	EB	Eingangsbyte	BYTE, CHAR	0..65535
IW	EW	Eingangswort	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Eingangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Ausgangsbit	BOOL	0.0..65535.7
QB	AB	Ausgangsbyte	BYTE, CHAR	0..65535
QW	AW	Ausgangswort	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Ausgangsdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Merkerbit	BOOL	0.0..65535.7
MB	MB	Merkerbyte	BYTE, CHAR	0..65535
MW	MW	Merkerwort	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Merkerdoppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Peripherieeingangs-byte	BYTE, CHAR	0..65535
PQB	PAB	Peripherieausgangs-byte	BYTE, CHAR	0..65535
PIW	PEW	Peripherieeingangs-wort	WORD, INT, S5TIME, DATE	0..65534
PQW	PAW	Peripherieausgangs-wort	WORD, INT, S5TIME, DATE	0..65534
PID	PED	Peripherieeingangs-doppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
PQD	PAD	Peripherieausgangs-doppelwort	DWORD, DINT, REAL, TOD, TIME	0..65532
T	T	Zeit	TIMER	0..65535
C	Z	Zähler	COUNTER	0..65535
FB	FB	Funktionsbaustein	FB	0..65535
OB	OB	Organisationsbaustein	OB	1..65535
DB	DB	Datenbaustein	DB, FB, SFB, UDT	1..65535
FC	FC	Funktion	FC	0..65535
SFB	SFB	Systemfunktions-baustein	SFB	0..65535
SFC	SFC	Systemfunktion	SFC	0..65535
VAT	VAT	Variablentabelle		0..65535
UDT	UDT	Anwenderdefinierter Datentyp	UDT	0..65535

7.5.4 Unvollständige und mehrdeutige Symbole in der Symboltabelle

Unvollständige Symbole

Sie haben die Möglichkeit, auch unvollständige Symbole abzuspeichern. Dadurch können Sie beispielsweise zunächst nur den Namen festlegen und die Adreßangaben zu einem späteren Zeitpunkt nachtragen. Insbesondere können Sie die Arbeiten an der Symboltabelle zu einem beliebigen Zeitpunkt unterbrechen und den Zwischenstand speichern. Um das Symbol bei der Software-Erstellung (ohne Fehlermeldung) benutzen zu können, müssen jedoch die Bezeichnung des Symbols, die Adresse und der Datentyp eingetragen sein.

Entstehung mehrdeutiger Symbole

Mehrdeutige Symbole entstehen, wenn Sie ein Symbol in die Symboltabelle einfügen, dessen Name (Symbol) und/oder Adresse bereits in einem anderen Symbol vorhanden sind. Dadurch werden sowohl das ursprünglich vorhandene und das eingefügte Symbol mehrdeutig.

Dieser Fall tritt z. B. ein, wenn Sie ein Symbol kopieren und einfügen, um den Eintrag in der Kopie danach leicht zu modifizieren.

Kennzeichnung mehrdeutiger Symbole

In der Symboltabelle sind mehrdeutige Symbole durch grafische Hervorhebung (Farbe, Schriftart) gekennzeichnet. Die veränderte Darstellung weist Sie auf eine erforderliche Nachbearbeitung hin. Sie können alle Symbole anzeigen oder die Anzeige so filtern, daß nur eindeutige oder nur mehrdeutige Symbole angezeigt werden.

Beseitigung der Mehrdeutigkeit

Ein mehrdeutiges Symbol wird eindeutig, wenn Sie die Komponente (Name und/oder Adresse) ändern, die die Mehrdeutigkeit verursacht hat. Sind zwei Symbole mehrdeutig, und Sie machen ein Symbol eindeutig, so wird das andere ebenfalls wieder eindeutig.

7.6 Eingabemöglichkeiten von globalen Symbolen

7.6.1 Eingabemöglichkeiten von globalen Symbolen

Für die Eingaben von Symbolen, die bei der späteren Programmierung weiterverwendet werden, gibt es drei Möglichkeiten:

- Eingabe über Dialogfeld
  Sie können in einem Fenster zur Programmeingabe ein Dialogfeld öffnen und darin ein neues Symbol definieren. Das Vorgehen eignet sich zur Definition einzelner Symbole, beispielsweise wenn Sie während der Programmierung feststellen, daß ein Symbol fehlt oder korrigiert werden muß. Sie ersparen sich so das Anzeigen der Symboltabelle.
- Eingabe direkt in Symboltabelle
  Sie können Symbole und ihre zugeordneten Adressen direkt in eine ”Symboltabelle” eintragen. Dieses Verfahren empfiehlt sich zur Eingabe mehrerer Symbole und zur Ersterstellung der Symboltabelle, weil Sie die bereits definierten Symbole am Bildschirm angezeigt bekommen und so leichter die Übersicht behalten.
- Importieren von Symboltabellen aus anderen Tabelleneditoren
  Sie können die Daten für die Symboltabelle auch mit dem von ihnen bevorzugten Tabelleneditor (z. B. Microsoft Excel) erstellen und die erzeugte Datei dann in die Symboltabelle importieren.

7.6.2 Allgemeine Hinweise zur Eingabe von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Zeilen können Sie mit dem Menübefehl **Einfügen > Symbol** in die Symboltabelle vor der aktuellen Zeile einfügen. Wenn die Zeile vor der Cursor-Position bereits einen Operanden enthält, werden Sie beim Einfügen von neuen Symbolen unterstützt durch eine Vorbelegung der Spalten "Adresse" und "Datentyp". Die Adresse wird abgeleitet aus der vorhergehenden Zeile, als Datentyp wird der Default-Datentyp eingetragen.

Vorhandene Einträge können Sie mit Befehlen im Menü "Bearbeiten" kopieren und anschließend ändern. Speichern und schließen Sie dann die Symboltabelle. Sie können auch Symbole speichern, die noch nicht vollständig definiert sind.

Bei der Eingabe der Symbole in der Tabelle sind folgende Besonderheiten zu beachten:

Spalte	Hinweis
Symbol	Der Name muß innerhalb der gesamten Symboltabelle eindeutig sein. Wenn Sie die Eingabe in diesem Feld übernehmen oder das Feld verlassen, wird ein mehrdeutiges Symbol gekennzeichnet. Das Symbol darf maximal 24 Zeichen lang sein. Es sind keine Anführungszeichen " zugelassen.
Adresse	Wenn Sie die Eingabe in diesem Feld übernehmen oder das Feld verlassen, wird geprüft, ob die eingegebene Adresse zulässig ist.
Datentyp	Nach Eingabe der Adresse wird dieses Feld mit einer Default-Einstellung vorbelegt. Wenn Sie diese Einstellung ändern, wird geprüft, ob der neue Datentyp zur Adresse paßt.
Kommentar	Hier können Sie Anmerkungen eingeben, um die Funktionen der Symbole kurz zu erklären (max. 80 Zeichen). Die Angabe des Kommentars ist optional.

7.6.3 Eingeben einzelner globaler Symbole im Dialog

Die hier beschriebene Vorgehensweise zeigt Ihnen, wie Sie beim Programmieren von Bausteinen Symbole über Dialogfelder ändern oder neu definieren können, ohne daß dabei die Symboltabelle angezeigt werden muß.

Die Vorgehensweise ist nützlich, wenn Sie nur ein einzelnes Symbol bearbeiten wollen. Um mehrere Symbole zu ändern, sollten Sie die Symboltabelle öffnen und direkt in der Symboltabelle arbeiten.

Aktivieren der Anzeige von Symbolen im Baustein

Aktivieren Sie bei geöffnetem Baustein die Anzeige der Symbole im Bausteinfenster mit dem Menübefehl **Ansicht > Symbolische Darstellung**. Ein Häkchen vor dem Menübefehl kennzeichnet, daß die symbolische Darstellung aktiv ist.

Definieren von Symbolen bei der Programmeingabe

1. Stellen Sie sicher, daß im Bausteinfenster die symbolische Darstellung eingeschaltet ist (Menübefehl **Ansicht > Symbolische Darstellung**).
2. Markieren Sie im Anweisungsteil Ihres Programms die absolute Adresse, der Sie ein Symbol zuweisen wollen.
3. Wählen Sie den Menübefehl **Bearbeiten > Symbol**.
4. Füllen Sie das angezeigte Dialogfeld aus. Tragen Sie insbesondere ein Symbol ein und schließen Sie den Dialog ab.

Das definierte Symbol wird in die Symboltabelle eingetragen. Angaben, die zu mehrdeutigen Symbolen führen würden, werden mit Fehlermeldung abgewiesen.

Editieren in der Symboltabelle

Über den Menübefehl **Extras > Symboltabelle** können Sie die Symboltabelle zum Bearbeiten öffnen.

7.6.4 Eingeben mehrerer globaler Symbole in der Symboltabelle

Symboltabelle öffnen

Sie haben mehrere Möglichkeiten zum Öffnen einer Symboltabelle:

- Doppelklick auf die Symboltabelle im Projektfenster,
- Markieren der Symboltabelle im Projektfenster und Menübefehl **Bearbeiten > Objekt öffnen**.

Die Symboltabelle zum aktuellen Programm wird in einem eigenen Fenster angezeigt. Nun können Sie Symbole erzeugen oder ändern. Beim ersten Öffnen nach dem Erzeugen der Symboltabelle ist die Symboltabelle noch leer.

Eingeben von Symbolen

Um neue Symbole in die Symboltabelle einzutragen, navigieren Sie in die erste leere Zeile der Tabelle und füllen dann die Felder aus. Neue Leerzeilen können Sie mit dem Menübefehl **Einfügen > Symbol** in die Symboltabelle vor der aktuellen Zeile einfügen. Vorhandene Einträge können Sie mit Befehlen im Menü "Bearbeiten" kopieren und anschließend ändern. Speichern und schließen Sie dann die Symboltabelle. Sie können auch Symbole speichern, die noch nicht vollständig definiert sind.

Sortieren von Symbolen

Die Datensätze in der Symboltabelle lassen sich alphabetisch nach Symbol, Adresse, Datentyp oder Kommentar sortieren.

Die Sortierung kann z.B. in dem Dialogfeld geändert werden, das Sie mit dem Menübefehl **Ansicht > Sortieren...** aufrufen können.

Filtern von Symbolen

Mit einem Filter wählen Sie aus allen Datensätzen einer Symboltabelle eine Teilmenge aus.

Über den Menübefehl **Ansicht > Filtern** öffnen Sie das Dialogfeld "Filtern".

Hier können Sie Kriterien definieren, die die Datensätze erfüllen müssen, um herausgefiltert zu werden. Sie können filtern nach

- Namen, Adressen, Datentypen, Kommentaren
- Symbolen mit BuB-Atribut, Symbolen mit Kommunikationseigenschaften, Symbolen für Binärvariable bei Meldungen (Merker oder Prozeßeingang)
- Symbolen mit Status "gültig", "ungültig (mehrdeutig, unvollständig)"

Die einzelnen Kriterien sind mit UND verknüpft. Die herausgefilterten Datensätze beginnen mit den angegebenen Zeichenfolgen.

Wenn Sie mehr über die Möglichkeiten in dem Dialogfeld "Filtern" wissen möchten, öffnen Sie die Online-Hilfe durch Betätigen der Funktionstaste F1.

7.6.5 Groß-/Kleinschreibung bei Symbolen

Keine Unterscheidung mehr zwischen Groß- und Kleinschreibung

Bisher konnten in STEP 7 Symbole definiert werden, die sich nur durch die Groß- und Kleinschreibung einzelner Zeichen unterschieden. Ab STEP 7 V4.02 wurde dieses Verhalten geändert. Eine Unterscheidung der Symbole nach Groß- und Kleinschreibung ist ab dieser Version nicht mehr möglich.
Mit dieser Änderung wurde den Wünschen unserer Kunden Rechnung getragen, da damit mögliche Fehlerquellen in einem Programm deutlich reduziert werden. Mit der jetzt vorgenommen Einschränkung bei der Symboldefiniton werden ebenfalls die Ziele der PLCopen für die Definition eines Standards für übertragbare Programme unterstützt.

Eine unterschiedliche Symboldefinition, die sich nur aus der verwendeten Groß- oder Kleinschrift ergibt, wird zukünftig nicht mehr unterstützt. Bisher war z.B. folgende Definition in der Symboltabelle möglich:

Motor1 = E 0.0

motor1 = E 1.0

Die Symbole wurden aufgrund der Schreibweise (Groß-/Kleinschreibung) des ersten Buchstabens unterschieden. Bei dieser Art der Unterscheidung besteht eine große Verwechselungsgefahr. Mit der jetzigen Definition wurde diese mögliche Fehlerquelle ausgeschlossen.

Auswirkung auf bestehende Programme

Wenn Sie bei der Symboldefinition bisher dieses Kriterium zur Unterscheidung genutzt haben, kann es mit der neuen Definition zu Konflikten kommen wenn sich:

- Symbole **nur** durch Groß-/Kleinschreibung unterscheiden
- Parameter **nur** durch Groß-/Kleinschreibung unterscheiden
- Symbole **nur** durch Groß-/Kleinschreibung von Parametern unterscheiden

Alle drei Fälle können jedoch wie im Anschluß beschrieben analysiert und bereinigt werden.

Symbole, die sich nur durch Groß-/Kleinschreibung unterscheiden

Konflikt:

Falls die Symboltabelle noch nicht mit der aktuellen Softwareversion bearbeitet wurde, so wird beim Übersetzen von Quelldateien das oberste der mehrdeutigen Symbole aus der Symboltabelle verwendet.

Wurde die Symboltabelle bereits bearbeitet, so sind solche Symbole ungültig, d.h. beim Öffnen von Bausteinen wird keine Symbolik angezeigt und Quelldateien, die diese Symbole benutzen, sind nicht mehr fehlerfrei übersetzbar.

Abhilfe:

Überprüfen Sie ihre Symboltabelle auf Konfliktfälle indem Sie die Symboltabelle öffnen und neu sichern. Durch diese Aktion werden die mehrdeutigen Symbole erkannt. Sie können sich die mehrdeutigen Symbole nun über den Filter ”mehrdeutige Symbole” anzeigen lassen und diese korrigieren. Korrigieren Sie anschließend die Quelldateien mit Konflikten. Bei Bausteinen ist keine weitere Änderung nötig, da beim Öffnen automatisch die aktuelle (nun konfliktfreie) Symboltabelle verwendet bzw. angezeigt wird.

Parameter, die sich nur durch Groß-/Kleinschreibung unterscheiden

Konflikt:

Quelldateien, die solche Schnittstellen beinhalten, sind nicht mehr fehlerfrei übersetzbar. Bausteine mit solchen Schnittstellen können zwar noch geöffnet werden, jedoch ist kein Zugriff auf den zweiten dieser Parameter mehr möglich. Zugriffe auf den zweiten der Parameter werden beim Speichern automatisch auf den ersten Parameter umgesetzt.

Abhilfe:

Zur Überprüfung, welche Bausteine solche Konflikte enthalten, empfiehlt es sich, mit der Funktion ”Quelle generieren” für alle Bausteine eines Programms eine Quelldatei zu generieren. Wenn beim Versuch, die erzeugte Quelldatei neu zu übersetzen, Fehler auftreten, so liegt ein Konfliktfall vor.

Korrigieren Sie Ihre Quelldateien indem Sie die Parameter z.B. mittels der Funktion Suchen/Ersetzen eindeutig machen und übersetzen Sie diese neu.

Symbole, die sich nur durch Groß-/Kleinschreibung von Parametern unterscheiden

Konflikt:

Unterscheiden sich globale und lokale Symbole einer Quelldatei nur durch Groß-/Kleinschreibung und wurden keine Einleitungszeichen zur Kennzeichnung von globaler (”Symbolname”) bzw. lokaler (#Symbolname) Symbolik verwendet, so wird bei der Compilierung immer das lokale Symbol verwendet. Das führt zu einem veränderten Maschinencode.

Abhilfe:

In diesem Fall empfiehlt es sich, aus allen Bausteinen eine Quelle neu zu generieren. Dadurch werden automatisch die lokalen und globalen Zugriffe mit den entsprechenden Einleitungszeichen versehen und bei anschließenden Übersetzungsvorgängen korrekt behandelt.

7.6.6 Exportieren und Importieren von Symboltabellen

Sie können die angezeigte Symboltabelle in eine Textdatei exportieren, um Sie beispielsweise mit einem beliebigen Texteditor weiterzubearbeiten.

Sie können Tabellen, die Sie mit einem anderen Werkzeug erzeugt haben, in ihre Symboltabelle importieren und hier weiterbearbeiten. Die Import-Funktion läßt sich beispielsweise benutzen, um unter STEP5/ST erzeugte Zuordnungslisten nach Konvertierung in die Symboltabelle aufzunehmen.

Zur Auswahl stehen die Dateiformate *.SDF, *.ASC, *.DIF und *.SEQ.

Regeln zum Exportieren

Sie können die gesamte Symboltabelle, eine durch Filter begrenzte Teilmenge oder in der Tabellendarstellung markierte Zeilen exportieren.

Die Eigenschaften von Symbolen, die Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften...** einstellen können, werden nicht exportiert.

Regeln zum Importieren

- Für häufig verwendete Systemfunktionsbausteine (SFBs), Systemfunktionen (SFCs)und Organisationsbausteine (OBs)sind in der Datei ...\S7DATA\SYMBOL\SYMBOL.SDF Symbole vordefiniert, die Sie bei Bedarf importieren können.
- Eigenschaften von Symbolen, die Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften...** einstellen können, werden beim Exportieren und Importieren nicht berücksichtigt.

7.6.7 Dateiformate für den Import/Export einer Symboltabelle

Folgende Dateiformate können in die Symboltabelle importiert bzw. aus ihr exportiert werden:

- ASCII-Dateiformat (ASC)
- Data Interchange Format (DIF)
  DIF-Dateien können Sie in Microsoft Excel öffnen, bearbeiten und wieder speichern.
- System Data Format (SDF)
  SDF-Dateien können Sie in Microsoft Access öffnen, bearbeiten und wieder speichern.
  - Benutzen Sie für den Import und Export von Daten der Microsoft-Applikation ACCESS das SDF-Format.
  - Wählen Sie in ACCESS als Dateiformat ”Text (mit Trennzeichen)”.
  - Benutzen Sie als Texttrennzeichen das doppelte Hochkomma (”).
  - Benutzen Sie als Feldtrennzeichen das Komma (,).
- Zuordnungsliste (SEQ)
  Achtung: Beim Exportieren der Symboltabelle in eine Datei vom Typ .SEQ werden Kommentare, die länger als 40 Zeichen sind, nach dem 40. Zeichen abgeschnitten !

ASCII-Dateiformat (ASC)

Dateityp:	***.ASC**
Aufbau:	Datensatzlänge, Trennzeichen Komma, Datensatz
Beispiel:	126,grün_phase_fußg T 2 TIMER Dauer der Grünphase für Fußgänger 126,rot_fußg A 0.0 BOOL Rot für Fußgänger

Data Interchange Format (DIF)

Dateityp:	***.DIF**
Aufbau:	Eine DIF-Datei gliedert sich in Dateikopf (Header) und Daten:

Header	**TABLE**	**Beginn einer DIF-Datei**
	0,1	
	"<Titel>"	Kommentierende Zeichenkette
	VECTORS	Anzahl der Datensätze in der Datei
	0,<Datensatzzahl>	
	""	
	TUPLES	Anzahl der Datenfelder eines Datensatzes
	0,<Spaltenanzahl>	

Header	TABLE	Beginn einer DIF-Datei
	""	
	DATA	Kennung für Header-Ende und Datenbeginn
	0,0	
	""	
Daten (je Datensatz)	<Typ>,<numerischer Wert>	Kennung für den Datentyp, numerischer Wert
	<Zeichenkette>	Alphanumerischer Teil oder
	V	Falls alphanumerischer Teil unbenutzt

Header: Der Dateikopf muß die Satztypen TABLE, VECTORS, TUPLES und DATA in der angegebenen Reihenfolge enthalten. Vor DATA können in DIF-Dateien noch weitere, optionale Satztypen stehen. Diese werden vom Symbol Editor jedoch ignoriert.

Daten: Im Datenteil besteht jeder Eintrag aus 3 Teilen: der Kennung für den Typ, einem numerischen Wert und einem alphanumerischen Teil.

DIF-Dateien können Sie in Microsoft Excel öffnen, bearbeiten und wieder speichern. Sie sollten dabei allerdings auf die Verwendung von deutschen Umlauten und anderen sprachabhängigen Sonderzeichen verzichten.

System Data Format (SDF)

Dateityp:	*.SDF
Aufbau:	Zeichenketten in Hochkommata, Teile durch Kommata getrennt
Beispiel:	"grün_phase_fußg","T 2","TIMER","Dauer der Grünphase für Fußgänger" "rot_fußg","A 0.0","BOOL","Rot für Fußgänger"

Zum Öffnen einer SDF-Datei in Microsoft Access wählen Sie das Dateiformat "Text (mit Trennzeichen)". Geben Sie als Texttrennzeichen das doppelte Hochkomma (") und als Feldtrennzeichen das Komma (,) an.

Zuordnungsliste (SEQ)

Dateityp:	*.SEQ
Aufbau:	TAB Adresse TAB Symbol TAB Kommentar CR
Beispiel:	T 2 grün_phase_fußg Dauer der Grünphase für Fußgänger A 0.0 rot_fußg Rot für Fußgänger

TAB steht für das Tabulatorzeichen (09H),
CR für einen Zeilenumbruch mit der RETURN-Taste (0DH).

8 Anlegen von Bausteinen und Bibliotheken

8.1 Auswahl der Erstellmethode

Je nach Programmiersprache, die Sie für die Programmerstellung benutzen, können Sie Ihr Programm inkrementell und/oder quellorientiert eingeben.

Inkrementelle Editoren für die Programmiersprachen KOP, FUP, AWL und S7-GRAPH

Mit den inkrementellen Editoren für KOP, FUP, AWL und S7-GRAPH erstellen Sie Bausteine, die im Anwenderprogramm abgelegt werden. Entscheiden Sie sich für die inkrementelle Eingabe, wenn Sie Ihre Eingaben sofort kontrollieren lassen möchten. Diese Eingabeart kommt auch Programmieranfängern entgegen. Bei der inkrementellen Eingabe wird jede Zeile bzw. jedes Element sofort auf syntaktische Korrektheit überprüft. Eventuell vorhandene Fehler werden angezeigt und müssen vor dem Beenden der Eingabe beseitigt werden. Syntaktisch korrekte Eingaben werden automatisch übersetzt und im Anwenderprogramm abgelegt.

Verwendete Symbole müssen vor dem Editieren der Anweisungen definiert sein. Falls bestimmte Symbole nicht vorhanden sind, kann der Baustein nicht vollständig übersetzt werden; dieser ”inkonsistente Zwischenstand” kann jedoch abgespeichert werden.

Quellorientierte (Text-) Editoren für die Programmiersprachen AWL, S7-SCL oder S7-HiGraph

In quellorientierten Editoren erstellen Sie **Quellen**, die anschließend in Bausteine übersetzt werden.

Wählen Sie die quellorientierte Eingabe, um ein Programm schnell einzugeben oder zu erfassen.

Bei der quellorientierten Eingabe wird das Programm bzw. ein Baustein in einer Textdatei editiert und die Textdatei anschließend kompiliert

Die Textdateien (Quellen) werden im Quellordner Ihres S7-Programms z.B. als **AWL-Quelle** bzw. **SCL-Quelle** abgelegt. Eine Quelldatei kann dabei den Code für einen oder mehrere Bausteine enthalten. Mit den Texteditoren für AWL bzw. SCL können Sie Code für **OBs, FBs, FCs, DBs und UDTs** (anwenderdefinierte Datentypen)**,** also auch für ein komplettes Anwenderprogramm erstellen. Das gesamte Programm einer CPU (d.h. alle Bausteine) kann in einer einzigen Textdatei stehen.

Durch Übersetzung der Quelldatei werden die entsprechenden Bausteine erzeugt und im Anwenderprogramm abgelegt. Verwendete Symbole müssen vor dem Übersetzen definiert sein. Eventuelle Fehler werden erst bei der Übersetzung durch den entsprechenden Compiler gemeldet.

Für die Übersetzung ist es wichtig, daß Sie die vorgeschriebene Syntax der Programmiersprache berücksichtigen. Eine Syntaxprüfung erfolgt erst durch Ihren Befehl zur Konsistenzprüfung oder bei der Übersetzung in Bausteine.

8.2 Auswahl der Programmiersprache

8.2.1 Auswahl der Programmiersprache

Festlegen der Programmiersprache des Editors

In welcher Programmiersprache und welchem Editor Sie einen Baustein bzw. eine Quelldatei erstellen wollen, legen Sie bei der Erzeugung des Bausteins bzw. der Quelle in den Objekteigenschaften fest. Diesem Eintrag gemäß wird der entsprechende Editor beim Öffnen des Bausteins/der Quelldatei gestartet.

Spracheditor starten

Den jeweiligen Spracheditor starten Sie im SIMATIC Manager durch einen Doppelklick auf das entsprechende Objekt (Baustein, Quelldatei, etc.), über den Menübefehl **Bearbeiten > Objekt öffnen** oder über das entsprechende Symbol in der Funktionsleiste.

Zur Erstellung des S7-Programms stehen Ihnen die in der Tabelle angegebenen Programmiersprachen zur Verfügung. Die Programmiersprachen KOP, FUP und AWL sind in der Basissoftware STEP 7 enthalten. Die übrigen Programmiersprachen können Sie als Optionssoftware erwerben.

Damit haben Sie die Auswahl zwischen unterschiedlichen Programmierphilosophien (Kontaktplan, Funktionsplan, Anweisungsliste, Hochsprache, Ablaufsteuerung oder Zustandsgraph) und zwischen textueller oder graphischer Programmierung.

Durch die Wahl der Programmiersprache sind auch die zulässigen Eingabemethoden (•) festgelegt.

Programmier-sprache	**Anwendergruppe**	**Anwendungsfall**	**Inkremen-telle Eingabe**	**Quellorien-tierte Eingabe**	**Baustein aus der CPU rückdoku-mentierbar**
Anweisungsliste AWL	Anwender, die maschinennah programmieren wollen.	Laufzeit- und speicherplatz-optimierte Programme	•	•	•
Kontaktplan KOP	Anwender, die Umgang mit Stromlaufplänen gewohnt sind.	Programmierung von Verknüpfungs-steuerungen	•	–	•
Funktionsplan FUP	Anwender, die mit den logischen Boxen der Boole-schen Algebra vertraut sind.	Programmierung von Verknüpfungs-steuerungen	•	–	•
SCL (Structured Control Language) Optionspaket	Anwender, die in Hochsprachen wie PASCAL oder C programmiert haben.	Programmieren von Daten-verarbeitungs-aufgaben	–	•	–
GRAPH Optionspaket	Anwender, die technologieorien-tiert ohne tiefere Programmier-/SPS -Kenntnisse arbeiten wollen.	komfortable Beschreibung sequentieller Vorgänge	•	–	•
HiGraph Optionspaket	Anwender, die technologieorien-tiert ohne tiefere Programmier-/SPS -Kenntnisse arbeiten wollen.	komfortable Beschreibung asynchroner, nichtsequentieller Vorgänge.	–	•	–
CFC Optionspaket	Anwender, die technologieorien-tiert ohne tiefere Programmier-/SPS -Kenntnisse arbeiten wollen.	Beschreibung kontinuierlicher Vorgänge.	–	–	–

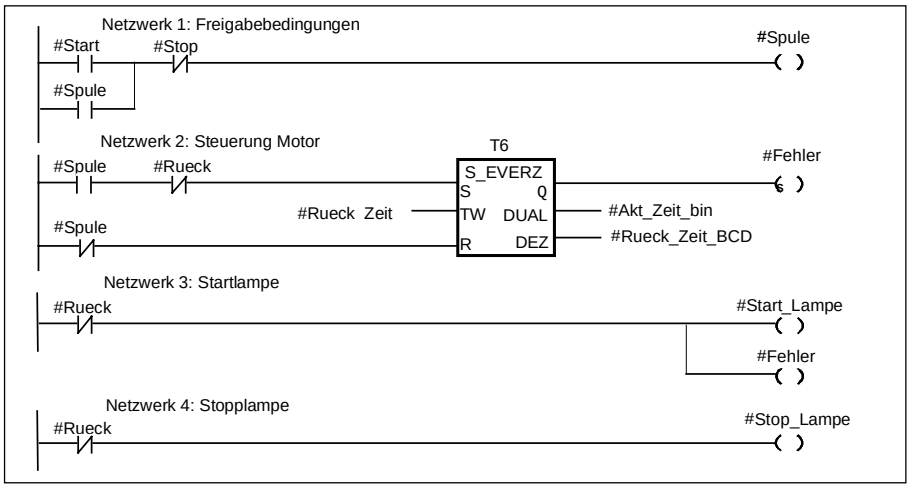
Bei fehlerfreien Bausteinen können Sie zwischen den Darstellungen des Bausteins in den Programmiersprachen KOP, FUP und AWL wechseln. Dabei werden Programmteile, die in der Zielsprache nicht darstellbar sind, in AWL dargestellt.

Bausteine können Sie aus Quelldateien in AWL erzeugen und hieraus auch wieder Quelldateien generieren.

8.2.2 Programmiersprache KOP (Kontaktplan)

Die Darstellung in der grafischen Programmiersprache KOP (Kontaktplan) ist Stromlaufplänen nachempfunden. Die Elemente eines Stromlaufplans, wie z. B. Öffner und Schließer, werden zu Netzwerken zusammengefügt. Ein oder mehrere Netzwerke ergeben den gesamten Anweisungsteil eines Codebausteins.

Beispiel für Netzwerke in KOP:



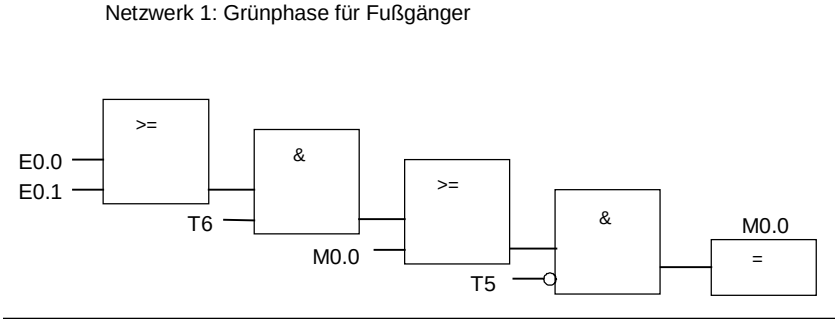
Die Programmiersprache KOP ist in der Basissoftware STEP 7 enthalten. Die Programmerstellung in KOP erfolgt mit einem inkrementellen Editor.

8.2.3 Programmiersprache FUP (Funktionsplan)

Die Programmiersprache FUP (Funktionsplan) benutzt zur Darstellung der Logik die von der Booleschen Algebra bekannten grafischen Logiksymbole. Außerdem können komplexe Funktionen, wie z.B. mathematische Funktionen, direkt in Verbindung mit den logischen Boxen dargestellt werden.

Die Programmiersprache FUP ist in der Basissoftware STEP 7 enthalten.

Beispiel für ein Netzwerk in FUP:



Die Programmerstellung in FUP erfolgt mit einem inkrementellen Editor.

8.2.4 Programmiersprache AWL (Anweisungsliste)

Die Programmiersprache AWL (Anweisungsliste) ist eine maschinennahe textuelle Sprache. Die einzelnen Anweisungen entsprechen den Arbeitsschritten, mit denen die CPU die Abarbeitung des Programms durchführt. Mehrere Anweisungen können zu Netzwerken zusammengefügt werden.

Beispiel für Netzwerke in AWL:

```
        Netzwerk 1: Steuerung Abflußventil
U(
O
O    #Spule
)
UN   #Schliessen
=    #Spule

        Netzwerk 2: Anzeige "Ventil offen"
U   #Spule
=    #Anz_offen

          Netzwerk 3: Anzeige "Ventil geschlossen"
UN   #Spule
=    #Anz_geschl
```

Die Programmiersprache AWL ist in der Basissoftware STEP 7 enthalten. Sie können in dieser Programmiersprache mit inkrementellen Editoren S7-Bausteine bearbeiten oder Ihr Programm mit einem quellorientierten Editor in einer AWL-Quelle erstellen und anschließend in Bausteine übersetzen.

8.2.5 Programmiersprache S7-SCL

Die als Optionssoftware erhältliche Programmiersprache SCL (Structured Control Language) ist eine textuelle Hochsprache und entspricht in ihrer Sprachdefinition im wesentlichen der IEC 1131-3. Die PASCAL-artige Sprache vereinfacht durch ihre Hochsprachenbefehle im Gegensatz zu AWL z.B. die Programmierung von Schleifen und bedingten Verzweigungen. SCL eignet sich daher u.a. für Formelberechnungen, komplexe Optimierungsalgorithmen oder die Verwaltung größerer Datenmengen.

Die Programmerstellung in S7-SCL erfolgt mit einem quellorientierten Editor in einer SCL-Quelle.

Beispiel:

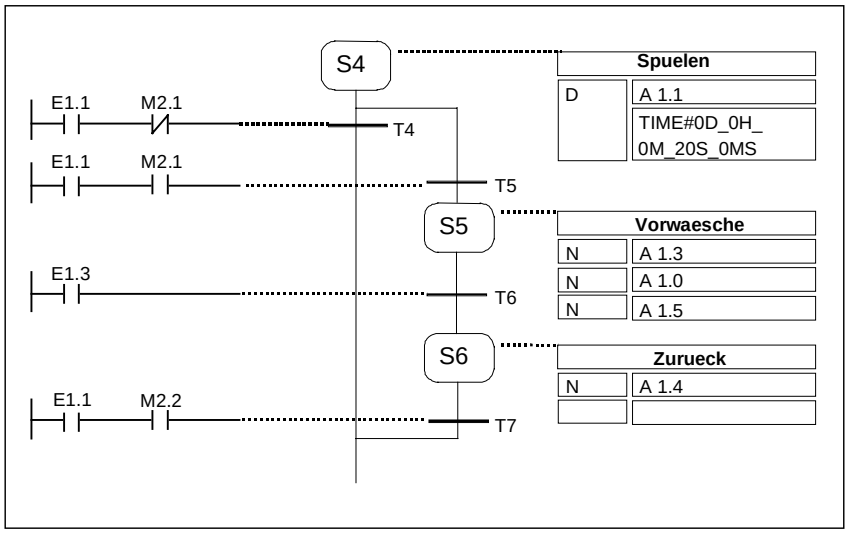
```
FUNCTION_BLOCK FB 20
VAR_INPUT
ENDWERT    :          INT;
END_VAR
VAR_IN_OUT
IQ1 :          REAL;
END_VAR
VAR
INDEX          :          INT;
END_VAR
BEGIN
CONTROL    := FALSE;
FOR INDEX := 1 TO ENDWERT DO
    IQ1 := IQ1 * 2;
    IF IQ1 > 10000 THEN
            CONTROL = TRUE
    END_IF;
END_FOR;
END_FUNCTION_BLOCK
```

8.2.6 Programmiersprache S7-GRAPH (Ablaufsteuerung)

Die als Optionssoftware erhältliche graphische Programmiersprache S7-GRAPH ermöglicht Ihnen die Programmierung von Ablaufsteuerungen. Dazu gehören die Erstellung einer Schrittkette, die Festlegung der jeweiligen Schrittinhalte und der Weiterschaltbedingungen (Transitionen). Den Schrittinhalt legen Sie in einer speziellen Programmiersprache (ähnlich AWL) fest, die Weiterschaltbedingungen werden in einer Kontaktplandarstellung eingegeben (eingeschränkter Umfang der Programmiersprache KOP).

S7-GRAPH stellt auch komplexe Abläufe sehr übersichtlich dar und erlaubt damit eine effektive Programmierung und Fehlersuche.

Beispiel für eine Ablaufsteuerung in S7-GRAPH



Erzeugte Bausteine

Mit dem Editor für S7-GRAPH programmieren Sie den Funktionsbaustein, der die Schrittkette beinhaltet. Die Daten für die Schrittkette, z. B. FB-Parameter, Schritt- und Transitionsbedingungen, enthält ein dazugehöriger Instanz-DB. Eine automatische Erzeugung dieses Instanz-DB veranlassen Sie im Editor für S7-GRAPH.

Quelldatei

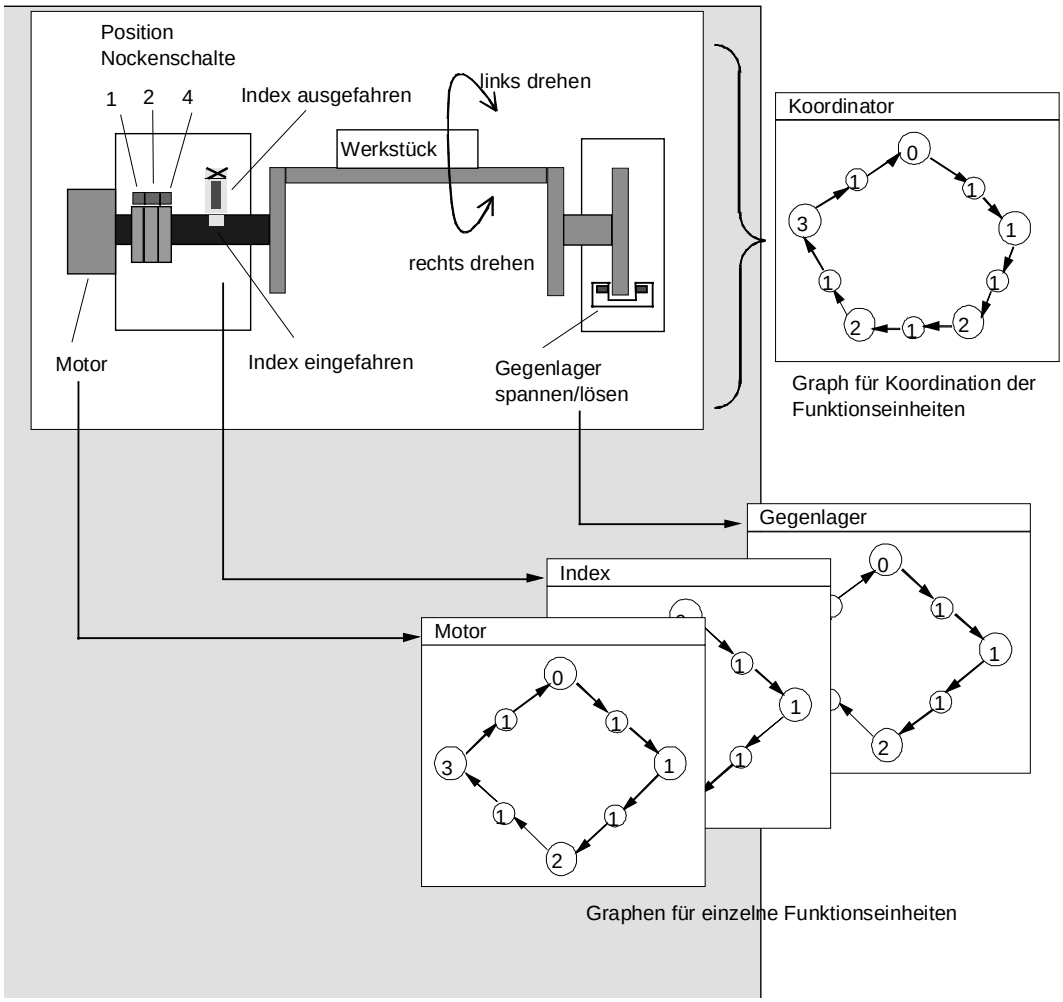
Aus einem mit S7-GRAPH erstellten FB kann eine textuelle Quelldatei (GRAPH-Quelle) erzeugt werden, die von Operator Panels oder Textdisplays zur Anzeige der Ablaufsteuerung interpretiert werden kann.

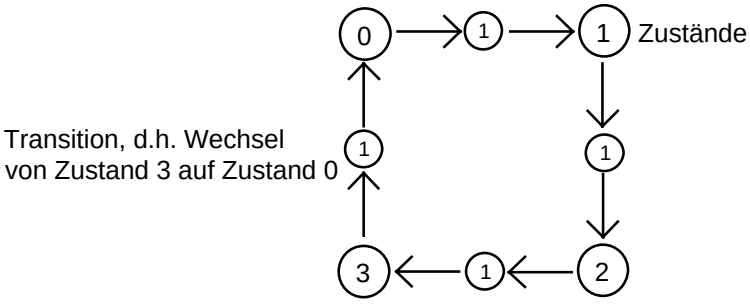
8.2.7 Programmiersprache S7-HiGraph (Zustandsgraph)

Die als Optionssoftware erhältliche grafische Programmiersprache S7-HiGraph ermöglicht Ihnen die Programmierung einiger Bausteine Ihres Programmes als Zustandsgraph. Dabei zerlegen Sie Ihre Anlage in eigenständige Funktionseinheiten, die jeweils verschiedene Zustände annehmen können. Für den Wechsel zwischen den Zuständen definieren Sie Transitionen. Die Aktionen, die den Zuständen zugeordnet sind, und die Bedingungen für die Transitionen zwischen den Zuständen beschreiben Sie in einer an AWL angelehnten Lupensprache.

Für jede Funktionseinheit erstellen Sie einen Graphen, der das Verhalten dieser Funktionseinheit beschreibt. Die Graphen einer Anlage werden zu Graphengruppen zusammengefaßt. Zwischen den Graphen können Nachrichten zur Synchronisation der Funktionseinheiten ausgetauscht werden.

Die übersichtliche Darstellung der Zustandsübergänge einer Funktionseinheit erlaubt eine systematische Programmierung und erleichtert die Fehlersuche. Im Unterschied zu S7-GRAPH ist bei S7-HiGraph zu einem Zeitpunkt immer nur ein Zustand (bei S7-GRAPH "Schritt") aktiv. Nachfolgendes Bild zeigt das Erstellen von Graphen für Funktionseinheiten (Beispiel).





Eine Graphengruppe wird in einer HiGraph-Quelle im Ordner "Quellen" unter dem S7-Programm abgespeichert. Diese Quelle wird dann in die S7-Bausteine für das Anwenderprogramm übersetzt.

Syntax und Formalparameter werden bei der letzten Eingabe für einen Graph (mit Schließen des Arbeitsfensters) geprüft. Die Adressen und Symbole werden erst beim Übersetzen der Quelle geprüft.

8.2.8 Programmiersprache S7-CFC

Die Optionssoftware CFC (*Continuous Function Chart*) ist eine Programmiersprache zur grafischen Verschaltung komplexer Funktionen.

Mit der Programmiersprache S7-CFC programmieren Sie, indem Sie vorhandene Funktionen graphisch verschalten. Viele Standardfunktionen brauchen Sie nicht selbst programmieren, sondern Sie können auf Bibliotheken mit Standardbausteinen (z. B. für Logik-, Arithmetik-, Regler- und Datenverarbeitungsfunktionen) zurückgreifen. Sie benötigen für die Anwendung von CFC keine tieferen Programmierkenntnisse oder spezifische SPS-Kenntnisse und können sich auf die branchenorientierte Technologie konzentrieren.

Das erstellte Programm wird in Form von CFC-Plänen gespeichert. Diese werden im Ordner "Pläne" unter dem S7-Programm abgelegt. Aus diesen Plänen werden dann die S7-Bausteine für das Anwenderprogramm übersetzt.

Zu verschaltende Bausteine, die Sie selbst erstellen möchten, programmieren Sie für SIMATIC S7 mit einer der S7-Programmiersprachen, für SIMATIC M7 mit C/C++.

8.3 Anlegen von Bausteinen

8.3.1 Baustein-Ordner

Das Programm für eine S7-CPU können Sie erstellen in Form von:

- Bausteinen
- Quellen

Für die Ablage von Bausteinen steht der Ordner "Bausteine" unter dem S7-Programm zur Verfügung.

Im Bausteinordner sind die Bausteine enthalten, die Sie für die Realisierung Ihrer Automatisierung auf die S7-CPU laden. Zu diesen ladbaren Bausteinen gehören Codebausteine (OBs, FBs, FCs) und Datenbausteine (DB). Ein leerer Codebaustein OB1 wird automatisch mit dem Ordner "Bausteine" angelegt, da Sie diesen Baustein auf jeden Fall für die Bearbeitung Ihres Programms in der S7-CPU benötigen.

Zusätzlich enthält der Ordner "Bausteine" noch folgende Objekte:

- Die von Ihnen erstellten anwenderdefinierten Datentypen. Sie vereinfachen Ihnen die Programmierung, werden aber nicht auf die CPU geladen.
- Die Variablentabellen (VATs), die Sie für den Test Ihres Programms zum Beobachten und Steuern von Variablen erstellen können. Variablentabellen werden nicht auf die CPU geladen.
- Das Objekt "Systemdaten" (Systemdatenbausteine), das Systeminformationen (Systemkonfiguration, Systemparameter) enthält. Diese Systemdatenbausteine werden durch das Konfigurieren der Hardware angelegt und mit Daten versorgt.
- Die Systemfunktionen (SFCs) und Systemfunktionsbausteine (SFBs), die Sie in Ihrem Anwenderprogramm aufrufen wollen. Die SFCs und SFBs selbst können Sie nicht bearbeiten.

Mit Ausnahme der Systemdatenbausteine (die nur über die Konfiguration des Automatisierungssystems erstellt und bearbeitet werden) erfolgt die Bearbeitung der Bausteine des Anwenderprogramms jeweils mit dem zugeordneten Editor. Dieser wird automatisch bei Doppelklick auf den entsprechenden Baustein gestartet.

Hinweis

Im Ordner "Bausteine" werden auch die Bausteine gespeichert, die Sie in Form von Quellen programmiert und anschließend übersetzt haben.

8.3.2 Anwenderdefinierte Datentypen (UDT)

Anwenderdefinierte Datentypen sind von Ihnen erzeugte spezielle Datenstrukturen, die Sie nach Ihrer Definition im gesamten S7-Anwenderprogramm verwenden können.

- UDTs können wie elementare Datentypen oder zusammengesetzte Datentypen in der Variablendeklaration von Codebausteinen (FC, FB, OB) oder als Datentyp für Variablen in einem Datenbaustein (DB) verwendet werden. Damit haben Sie den Vorteil, daß Sie eine mehrmals zu verwendende, spezielle Datenstruktur nur einmal definieren und sie anschließend beliebig vielen Variablen zuweisen können.
- UDTs können als Vorlage für die Erstellung von Datenbausteinen mit gleicher Datenstruktur dienen, d.h. Sie erstellen nur einmal die Struktur und erzeugen anschließend die benötigten Datenbausteine durch einfache Zuweisung des UDTs (Beispiel Rezepturen: Die Struktur des DB ist immer gleich, nur die Mengenangaben sind unterschiedlich).

Anwenderdefinierte Datentypen werden – wie die übrigen Bausteine auch – im SIMATIC Manager oder im inkrementellen Editor angelegt.

Aufbau eines UDT

Nach dem Öffnen eines UDT wird Ihnen in einem neuen Arbeitsfenster die Tabellendarstellung für diesen anwenderdefinierten Datentyp in Deklarationssicht angezeigt.

- Die erste und die letzte Zeile beinhalten bereits die Deklarationen STRUCT bzw. END_STRUCT für Beginn und Ende des anwenderdefinierten Datentyps. Diese Zeilen können Sie nicht editieren.
- Sie bearbeiten den anwenderdefinierten Datentyp, indem Sie Ihre Eingaben ab der zweiten Zeile der Deklarationstabelle in den entsprechenden Spalten durchführen.
- Sie können anwenderdefinierte Datentypen strukturieren aus:
  - elementaren Datentypen
  - zusammengesetzten Datentypen
  - bereits vorhandenen anwenderdefinierten Datentypen

Die anwenderdefinierten Datentypen des S7-Anwenderprogramms werden nicht auf die S7-CPU geladen. Sie werden entweder direkt durch einen inkrementellen Editor erstellt und bearbeitet oder sie entstehen durch die Übersetzung von Quellen.

8.3.3 Bausteineigenschaften

Mit Hilfe von Bausteineigenschaften können Sie die erstellten Bausteine besser identifizieren (z.B. bei der Versionspflege) oder Sie schützen Bausteine vor unberechtigten Veränderungen.

Die Bausteineigenschaften sollten Sie bei geöffnetem Baustein bearbeiten. Neben editierbaren Eigenschaften werden in dem zugehörigen Dialogfeld auch Daten nur zu ihrer Information angezeigt: Diese können Sie nicht bearbeiten.

Die Bausteineigenschaften und Systemattribute werden zwar auch im SIMATIC Manager in den Objekteigenschaften für einen Baustein angezeigt. Dort können Sie jedoch nur die Eigenschaften NAME, FAMILIE, AUTHOR und VERSION editieren.

Die Objekteigenschaften editieren Sie nach dem Einfügen des Bausteins über den SIMATIC Manager. Wurde ein Baustein nicht im SIMATIC Manager, sondern über einen der Editoren erzeugt, so sind diese Eintragungen (z.B. Programmiersprache) automatisch in den Objekteigenschaften gespeichert.

Hinweis

Die Mnemonik, in der Sie Ihre S7-Bausteine programmieren wollen, legen Sie im SIMATIC Manager über den Menübefehl **Extras > Einstellungen** im Register "Sprache" fest.

Tabelle der Bausteineigenschaften

Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge der Eingabe nach folgender Tabelle zu beachten.

Schlüsselwort / Eigenschaft	Bedeutung	Beispiel
[KNOW_HOW_PROTECT]	Bausteinschutz; ein Baustein, der mit dieser Option übersetzt wurde, läßt keinen Einblick in den Anweisungsteil zu.	KNOW_HOW_PROTECT
[AUTHOR:]	Name des Autors, Firmenname, Abteilungsname oder andere Namen (max. 8 Zeichen, ohne Leerzeichen)	AUTHOR : Siemens, aber kein Schlüsselwort
[FAMILY:]	Name der Bausteinfamilie: z.B. Regler. (max. 8 Zeichen, ohne Leerzeichen).	FAMILY : Regler, aber kein Schlüsselwort
[NAME:]	Bausteinname (max. 8 Zeichen)	NAME : PID, aber kein Schlüsselwort
[VERSION: int1 . int2]	Versionsnummer des Bausteins (beide Zahlen zwischen 0..15, d.h. 0.0 - 15.15))	VERSION : 3.10
[CODE_VERSION1]	Kennung, ob ein FB multiinstanzfähig ist oder nicht. Wenn Sie Multi-instanzen deklarieren möchten, darf der FB diese Eigenschaft nicht haben.	CODE_VERSION1
[UNLINKED] nur für DBs!	Ein Datenbaustein mit der Eigenschaft UNLINKED wird nicht in das Programm eingebunden.	
[READ_ONLY] nur für DBs!	Schreibschutz für Datenbausteine; seine Daten können nur gelesen u. nicht verändert werden.	READ_ONLY

Der Bausteinschutz KNOW_HOW_PROTECT hat folgende Konsequenzen:

- Wenn Sie später einen übersetzten Baustein im inkrementellen AWL-, FUP- oder KOP-Editor ansehen wollen, kann der Anweisungsteil des Bausteins nicht eingesehen werden.
- In der Variablendeklarationsliste des Bausteins werden nur die Variablen der Deklarationstypen var_in, var_out und var_in_out angezeigt. Die Variablen der Deklarationstypen var_stat und var_temp bleiben verborgen.

Zuordnung Bausteineigenschaften zu Bausteinarten

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinarten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
NAME	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–

Die Eigenschaft KNOW_HOW_PROTECT kann beim Programmieren des Bausteins in einer Quelle eingestellt werden. Im Dialogfeld zu den Baustein-Eigenschaften wird sie angezeigt, ist aber nicht veränderbar.

8.3.4 Anzeige von Baustein-Längen

Baustein-Längen werden in der Einheit "Byte" angezeigt.

Anzeige in den Eigenschaften eines Baustein-Ordners

In den Eigenschaften des Baustein-Ordners werden in Offline-Ansicht folgende Längen angezeigt:

Größe (Summe aller Bausteine ohne Systemdaten) im Ladespeicher des Zielsystems

Größe (Summe aller Bausteine ohne Systemdaten) im Arbeitsspeicher des Zielsystems

Bausteinlängen auf dem Erstellsystem (PG/PC) werden in den Eigenschaften des Baustein-Behälters nicht angezeigt.

Anzeige in den Eigenschaften eines Bausteins

In den Eigenschaften eines Bausteins werden angezeigt:

Benötigte Anzahl an Lokaldaten: Größe der Lokaldaten in Byte

MC7: Größe des MC7-Codes in Byte bzw. Größe der DB-Nutzdaten

Größe im Ladespeicher des Zielsystems

Größe im Arbeitsspeicher des Zielsystems: wird nur bei bekannter Hardware-Zuordnung angezeigt.

Die Anzeigen sind unabhängig davon, ob der Baustein im Fenster einer Online-Ansicht oder einer Offline-Ansicht liegt.

Anzeige im SIMATIC Manager (Ansicht Details)

Ist ein Baustein-Behälter geöffnet und "Ansicht Details" eingestellt, so wird im Projektfenster der Arbeitsspeicherbedarf angezeigt, unabhängig davon, ob der Baustein-Behälter im Fenster einer Online-Ansicht oder einer Offline-Ansicht liegt.

Sie können die Bausteinlängen summieren, indem Sie alle relevanten Bausteine markieren. In diesem Fall wird die Summe für die markierten Bausteine in der Statuszeile des SIMATIC Managers angezeigt.

Für nicht auf das Zielsystem ladbare Bausteine (z. B. VAT's) wird keine Länge dargestellt.

Bausteinlängen auf dem Erstellsystem (PG/PC) werden in der Ansicht Details nicht angezeigt.

8.3.5 Umverdrahten

Folgende Bausteine und Operanden können umverdrahtet werden:

- Eingänge, Ausgänge
- Merker, Zeiten, Zähler
- Funktionen, Funktionsbausteine

Gehen Sie so vor:

1. Markieren Sie im SIMATIC Manager den Ordner "Bausteine", in dem sich die Bausteine befinden, die Sie umverdrahten möchten.
2. Wählen Sie den Menübefehl **Extras > Umverdrahten**.
3. Tragen Sie im angezeigten Dialogfeld "Umverdrahten" die gewünschten Ersetzungen (Alter Operand / Neuer Operand) in die Tabelle ein.
4. Wählen Sie die Option "Alle Zugriffe innerhalb der angegebene Operanden", wenn Sie Operandenbereiche (BYTE, WORD, DWORD) umverdrahten möchten.
   Beispiel: Sie geben als Operandenbereiche EW0 und EW4 an. Dann werden die Operanden E0.0 - E1.7 auf die Operanden E4.0 - E5.7 umverdrahtet. Operanden aus dem umverdrahteten Bereich (z.B. E0.1) können danach nicht mehr einzeln in die Tabelle eingetragen werden.
5. Betätigen Sie die Schaltfläche OK.

Damit starten Sie das Umverdrahten. Nach der Verdrahtung können Sie über ein Dialogfeld entscheiden, ob Sie die Infodatei zur Umverdrahtung ansehen wollen. Die Infodatei enthält die Operandenliste, "Alter Operand" und "Neuer Operand". Dazu werden die einzelnen Bausteine aufgeführt mit der Anzahl Verdrahtungen, welche jeweils im Baustein durchgeführt worden sind.

Beim Umverdrahten ist folgendes zu beachten:

- Wenn Sie einen Baustein umverdrahten (d. h. umbenennen), dann darf der neue Baustein nicht bereits existieren. Falls der Baustein bereits existiert, wird der Vorgang abgebrochen.
- Wenn Sie einen Funktionsbausteine (FB) umverdrahten, dann wird automatisch sein Instanz-DB dem umverdrahteten FB zugeordnet. Der Instanz-DB ändert sich nicht (d. h. die DB-Nummer bleibt erhalten)

8.3.6 Attribute für Bausteine und Parameter

Eine Beschreibung der Attribute finden Sie in der Referenzhilfe zu Systemattributen.

8.4 Arbeiten mit Bibliotheken

8.4.1 Arbeiten mit Bibliotheken

Bibliotheken dienen zur Ablage von wiederverwendbaren Programmkomponenten für die SIMATIC S7/M7. Die Programmkomponenten können aus vorhandenen Projekten in eine Bibliothek kopiert oder direkt in der Bibliothek unabhängig von Projekten erstellt werden.

Sie können sich viel Programmieraufwand sparen, wenn Sie Bausteine, die Sie immer wieder verwenden wollen, in einem S7-Programm unter einer Bibliothek ablegen. Von dort können Sie diese immer in das entsprechende Anwenderprogramm kopieren.

Für die Erstellung von S7/M7-Programmen in einer Bibliothek steht Ihnen - mit Ausnahme des Testens - die gleiche Funktionalität wie in Projekten zur Verfügung.

Anlegen von Bibliotheken

Bibliotheken können Sie wie Projekte mit dem Menübefehl **Datei > Neu** anlegen. Die neue Bibliothek wird in dem Verzeichnis angelegt, das Sie nach Anwahl des Menübefehls **Extras > Einstellungen** im Register “Allgemein” für die Bibliotheken eingestellt haben.

Hinweis

Der SIMATIC Manager läßt Namen zu, die länger als 8 Zeichen sind. Der Name des Bibliotheksverzeichnisses wird jedoch auf 8 Zeichen abgeschnitten. Bibliotheksnamen müssen sich daher in den ersten 8 Zeichen unterscheiden. Zwischen Groß- und Kleinschreibung wird nicht unterschieden. Beim Öffnen wird im Browser wieder der vollständige Name angezeigt, beim Durchsuchen nur der gekürzte.

Beachten Sie, daß Sie in Projekten einer älteren STEP 7-Version keine Bausteine aus Bibliotheken einer neueren STEP 7-Version verwenden können.

Öffnen von Bibliotheken

Um eine Bibliothek zu öffnen, geben Sie zunächst den Menübefehl **Datei > Öffnen** ein. Wählen Sie dann eine Bibliothek in den Folgedialogen aus. Danach wird das Bibliotheksfenster geöffnet.

Hinweis

Falls Sie die gewünschte Bibliothek nicht in der Bibliotheksliste angezeigt bekommen, betätigen Sie im Dialogfeld “Öffnen” die Schaltfläche “Durchsuchen”. Daraufhin wird Ihnen im Windows-Standarddialogfeld die Verzeichnisstruktur angezeigt, in der Sie die Bibliothek suchen können.

Beachten Sie hierbei, daß der Name der Datei immer dem ursprünglichen Namen der angelegten Bibliothek entspricht, d. h. Namensänderungen im SIMATIC Manager werden auf Dateiebene nicht nachgezogen.

Sobald Sie die Bibliothek ausgewählt haben, wird diese in die Bibliotheksliste aufgenommen. Die Einträge in der Bibliotheksliste können Sie mit dem Menübefehl **Datei > Verwalten** ändern.

Kopieren von Bibliotheken

Eine Bibliothek kopieren Sie, indem Sie diese über den Menübefehl **Datei > Speichern unter** unter einen anderen Namen abspeichern.

Bibliotheksteile wie Programme, Bausteine, Quellen usw. kopieren Sie mit dem Menübefehl **Bearbeiten > Kopieren**.

Löschen von Bibliothek

Eine Bibliothek löschen Sie über den Menübefehl **Datei > Löschen**.

Bibliotheksteile wie Programme, Bausteine, Quellen usw. löschen Sie mit dem Menübefehl **Bearbeiten > Löschen**.

8.4.2 Hierarchischer Aufbau von Bibliotheken

Ebenso wie Projekte sind Bibliotheken hierarchisch aufgebaut:

- Bibliotheken können S7/M7-Programme enthalten.
- Ein S7-Programm kann genau einen Ordner “Bausteine” (Anwenderprogramm), einen Ordner “Quellen”, einen Ordner “Pläne” sowie ein Objekt “Symbole” (Symboltabelle) enthalten.
- Ein M7-Programm kann Pläne und C-Programme für programmierbare M7-Baugruppen sowie ein Objekt “Symbole” (Symboltabelle) und einen Bausteinordner für DBs und VATs (Variablentabellen) enthalten.
- Der Ordner “Bausteine” enthält die Bausteine, die auf die S7-CPU geladen werden können. Die enthaltenen Variablentabellen und anwenderdefinierten Datentypen werden nicht auf die CPU geladen.
- Der Ordner “Quellen” enthält die Quellen für die in den diversen Programmiersprachen erstellten Programmen.
- Der Ordner “Pläne” enthält die CFC-Pläne (nur bei Optionssoftware S7-CFC).

Wenn Sie ein neues S7-/M7-Programm einfügen, wird darin automatisch je ein Ordner “Bausteine” und “Quellen” (nur bei S7) sowie ein Objekt “Symbole” eingefügt.

8.4.3 Übersicht der Standardbibliotheken

Das STEP 7-Basispaket enthält die Standardbibliotheken (Version 2/Version 3):

- stlibs (V2): Standardbibliothek Version 2
- Standard Library: Standardbibliothek ab Version 3

Die Standardbibliotheken enthalten folgende Komponenten:

- **Sytem Function Blocks:** Systemfunktionen (SFC) und Systemfunktionsbausteine (SFB)
- **S5-S7 Converting Blocks:** Bausteine für die Konvertierung von STEP 5-Programmen
- **TI-S7 Converting Blocks:** allgemein verwendbare Standardfunktionen
- **IEC Function Blocks:** Bausteine für IEC-Funktionen, wie z. B. für die Bearbeitung von Zeit- und Datumsangaben, für Vergleichsoperationen, für String-Bearbeitung und zur Auswahl von Maximum und Minimum
- **Organization Blocks:** Standard-Organisationsbausteine (OB)

Die Standardbibliothek für Version 3 enthält zusätzlich die Komponenten:

- **PID Control Blocks:** Funktionsbausteine (FB) zur PID-Regelung
- **Communication Blocks:** Funktionen (FC) und Funktionsbausteine (FB) für SIMATICNET-CPs.

Weitere Bibliotheken können bei der Installation von Optionssoftware dazukommen.

Löschen und Installieren von mitgelieferten Bibliotheken

Die mitgelieferten Bibliotheken können Sie im SIMATIC Manager löschen und danach wieder neu installieren. Für die Installation führen Sie das Programm Setup von STEP 7 V5.0 komplett neu aus.

Hinweis

Die mitgelieferten Bibliotheken werden bei einer STEP 7-Installation immer kopiert. Falls Sie mitgelieferte Bibliotheken bearbeitet haben, werden bei einer erneuten STEP 7- Installation die von Ihnen geänderten Bibliotheken mit dem Original überschrieben.

Deswegen sollten Sie mitgelieferte Bibliotheken vor Änderungen kopieren und nur die Kopie bearbeiten.

---

9 Erstellen von Codebausteinen

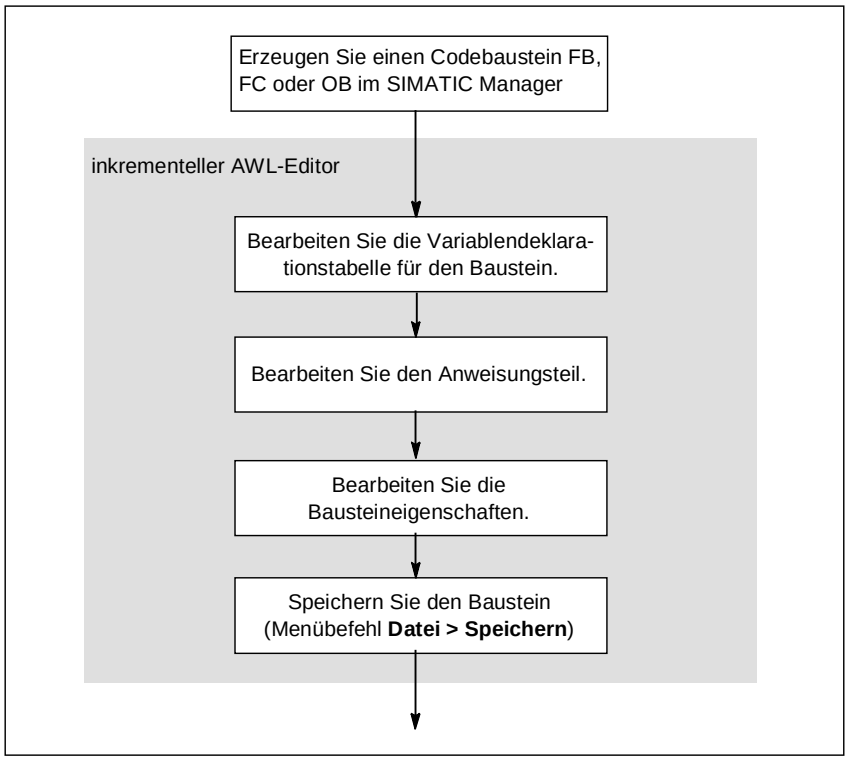
9.1 Grundlagen zum Erstellen von Codebausteinen

9.1.1 Prinzipielle Vorgehensweise beim Erstellen von Codebausteinen

Codebausteine (OBs, FBs, FCs) bestehen aus einem Variablendeklarationsteil, einem Anweisungsteil und besitzen zusätzlich Eigenschaften. Bei der Programmierung müssen Sie also folgende drei Teile bearbeiten:

- **Variablendeklarationstabelle**. In der Variablendeklarationstabelle legen Sie die Parameter, Systemattribute für Parameter sowie lokale Variablen des Bausteins fest.
- **Anweisungsteil**: Im Anweisungsteil programmieren Sie den Bausteincode, der vom Automatisierungssystem bearbeitet werden soll. Er besteht aus einem oder mehreren Netzwerken. Für die Erstellung der Netzwerke stehen Ihnen z.B. die Programmiersprachen AWL (Anweisungsliste), KOP (Kontaktplan) und FUP (Funktionsplan) zur Verfügung.
- **Bausteineigenschaften:** Die Bausteineigenschaften beinhalten zusätzliche Informationen wie Zeitstempel oder Pfadangabe, die vom System eingetragen werden. Daneben können Sie selbst Angaben zu Namen, Familie, Version und Autor machen und Sie können Systemattribute für Bausteine vergeben.

Prinzipiell ist es gleich, in welcher Reihenfolge Sie die Teile eines Codebausteins bearbeiten. Selbstverständlich können Sie auch korrigieren/ergänzen.


Erzeugen Sie einen Codebaustein FB, FC oder OB im SIMATIC Manager
inkrementeller AWL-Editor
Bearbeiten Sie die Variablendeklara-tionstabelle für den Baustein.
Bearbeiten Sie den Anweisungsteil.
Bearbeiten Sie die Bausteineigenschaften.
Speichern Sie den Baustein (Menübefehl **Datei > Speichern**)

Hinweis

Wenn Sie auf Symbole der Symboltabelle zurückgreifen wollen, sollten Sie diese zuerst auf Vollständigkeit überprüfen und ggf. ergänzen.

9.1.2 Voreinstellungen für Programmeditor KOP/FUP/AWL

Bevor Sie mit der Programmierung beginnen, sollten Sie die Einstellmöglichkeiten kennenlernen, um möglichst komfortabel und Ihren Gewohnheiten entsprechend arbeiten zu können.

Über den Menübefehl **Extras > Einstellungen** öffnen Sie einen Registerdialog. In den verschiedenen Registern können Sie Voreinstellungen für das Programmieren von Bausteinen einstellen, z. B. im register "Editor":

- die Schrift (Art und Größe) in Text und Tabellen
- ob Sie bei einem neuen Baustein zunächst Symbolik und Kommentar angezeigt bekommen möchten.

Die Einstellungen für Sprache, Kommentar und Symbolik können Sie aktuell während der Bearbeitung über Befehle des Menüs **Ansicht > ...** ändern.

Die Farbe, in der Markierungen z.B. von Netzwerken oder Anweisungszeilen dargestellt werden, ändern Sie im Register ”KOP”.

9.1.3 Zugriffsrechte auf Bausteine bzw. Quellen

Bei der Bearbeitung eines Projektes wird häufig eine gemeinsame Datenhaltung benutzt, dadurch kann es vorkommen, daß mehrere Bearbeiter auf den gleichen Baustein oder die gleich Datenquelle zugreifen wollen.

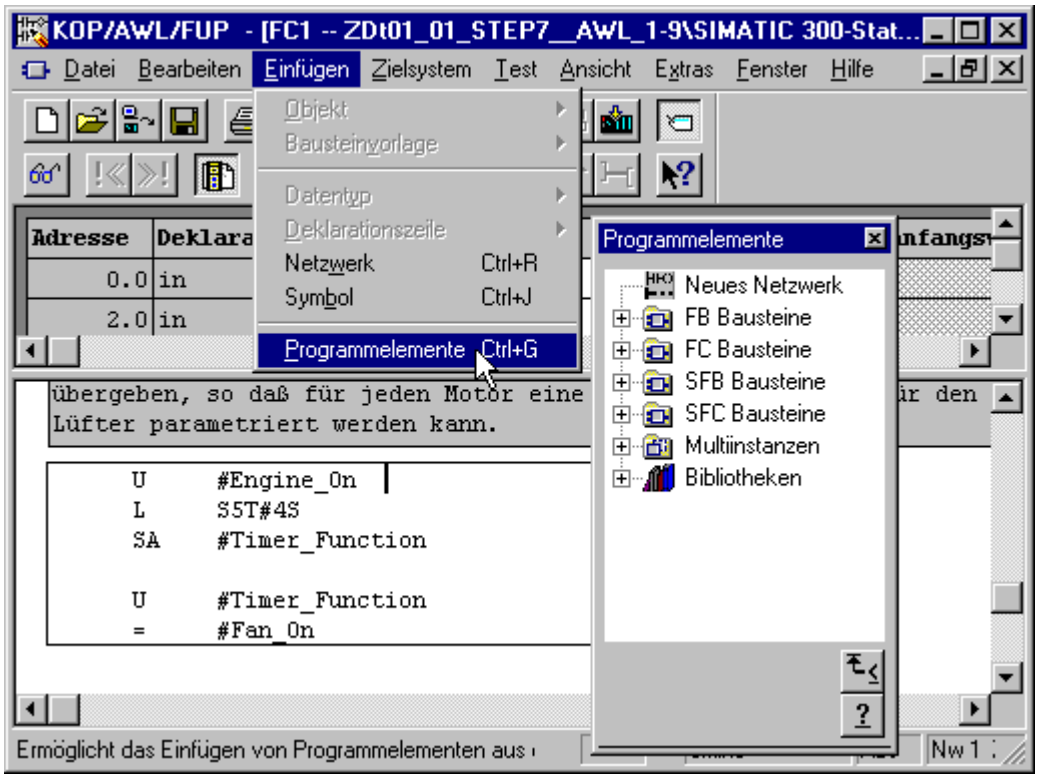
Die Schreib- / Leserechte sind wie folgt vergeben:

- Offline Bearbeitung:
  Beim Öffnen eines Bausteines/Quelle wird geprüft, ob dieser mit dem Recht 'Schreiben' geöffnet werden kann. Ist der Baustein/Quelle bereits geöffnet kann nur mit einer Kopie gearbeitet werden. Wollen Sie die Kopie speichern wird abgefragt ob Sie das Original überschreiben oder die Kopie unter eine anderen Namen speichern wollen.
- Online Bearbeitung:
  Öffnet der Bearbeiter über eine projektierte Verbindung einen Online-Baustein, so wird der zugehörige Offline-Baustein gesperrt und somit dessen gleichzeitige Änderung unterbunden

9.1.4 Anweisungen aus dem Programmelemente-Katalog

Der Katalog "Programmelemente stellt KOP- und FUP-Sprachelemente sowie bereits deklarierte Multiinstanzen, fertige Bausteine und Bausteine aus Bibliotheken zur Verfügung. Er läßt sich über den Menübefehl **Ansicht > Katalog** aufrufen. Programmelemente können mit dem Menübefehl **Einfügen > Programmelemente** in den Anweisungsteil übernommen werden.

Beispiel des Programmelemente-Katalogs bei AWL

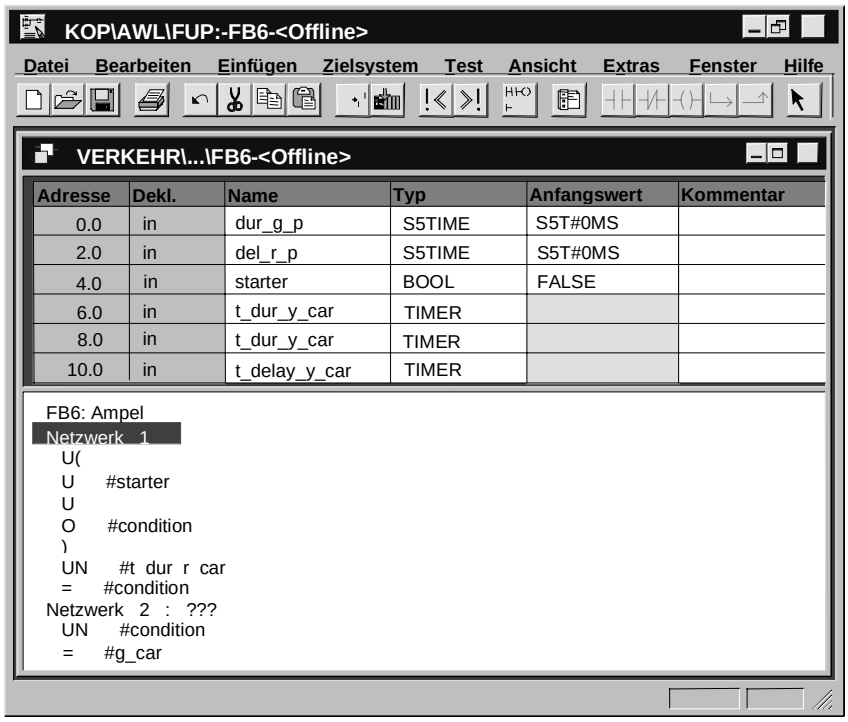


9.2 Editieren der Variablendeklarationstabelle

9.2.1 Verwendung der Variablendeklaration in Codebausteinen

Nach dem Öffnen eines Codebausteins erscheint ein Fenster mit der Variablendeklarationstabelle des Bausteins im oberen Teil und dem Anweisungsteil im unteren Teil, in dem Sie den eigentlichen Bausteincode bearbeiten.

Beispiel: Variablendeklarationstabelle und Anweisungsteil in AWL



In der Variablendeklarationstabelle legen Sie die lokalen Variablen einschliesslich der Formalparameter des Bausteins und der Systemattribute für Parameter fest. Das hat u.a. folgende Auswirkungen:

- Durch das Deklarieren wird entsprechend Speicherplatz reserviert, für temporäre Variablen im Lokaldaten-Stack, im Fall von Funktionsbausteinen, für statische Variablen im später zugeordneten Instanz-DB.
- Durch die Festlegung von Eingangs-, Ausgangs- und Durchgangsparametern legen Sie auch die ”Schnittstelle” für den Aufruf des Bausteins im Programm fest.
- Wenn Sie die Variablen in einem Funktionsbaustein deklarieren, bestimmen diese Variablen (mit Ausnahme der temporären) auch die Datenstruktur für jeden Instanz-DB, der dem FB zugeordnet wird.
- Durch die Festlegung von Systemattributen weisen Sie Parametern spezielle Eigenschaften für Meldungs- und Verbindungsprojektierung, Bedien- und Beobachtungsfunktionen und die Leittechnikprojektierung zu.

9.2.2 Zusammenspiel zwischen Variablendeklarationstabelle und Anweisungsteil

Variablendeklarationstabelle und Anweisungsteil von Codebausteinen sind eng miteinander verbunden, da im Anweisungsteil die Namen aus der Variablendeklarationstabelle verwendet werden. Änderungen in der Variablendeklaration wirken sich daher im gesamten Anweisungsteil aus.

Aktion in der Variablendeklaration	Reaktion im Anweisungsteil
Korrekte Neueingabe	Falls ungültiger Code vorhanden, wird zuvor nicht deklarierte Variable nun gültig.
Korrekte Namensänderung ohne Typänderung	Symbol wird sofort überall mit neuem Namen dargestellt.
Korrekter Name wird in einen ungültigen Namen geändert	Code wird nicht geändert.
Ungültiger Name wird in einen korrekten geändert	Falls ungültiger Code vorhanden, wird dieser gültig.
Typänderung	Falls ungültiger Code vorhanden, wird dieser gültig, und falls gültiger Code vorhanden, wird dieser unter Umständen ungültig.
Löschen einer Variable (Symbol), die im Code verwendet wird.	Gültiger Code wird ungültig

Kommentaränderungen, die fehlerhafte Eingabe einer neuen Variablen, eine Anfangswertänderung oder das Löschen einer nicht verwendeten Variablen haben keine Auswirkung auf den Anweisungsteil.

9.2.3 Aufbau der Variablendeklarationstabelle

Die Variablendeklarationstabelle beinhaltet Einträge für Adresse, Deklarationstyp, Name, Datentyp, Anfangswert und Kommentar der Variablen. Jede Tabellenzeile steht für eine Variablendeklaration. Variablen vom Datentyp ARRAY oder STRUCT benötigen mehrere Zeilen.

Zulässige Datentypen für die Lokaldaten der verschiedenen Bausteinarten finden sie in "Zuordnen von Datentypen zu Lokaldaten von Codebausteinen".

Spalte	Bedeutung	Anmerkungen	Bearbeitung
Adresse	Adresse im Format BYTE.BIT.	Bei Datentypen, für die mehr als ein Byte erforderlich ist, zeigt die Adresse die Zuordnung durch einen Sprung zur nächsten Byte-Adresse an. Zeichenlegende: * : Größe eines Feldelements in Byte. + : Anfangsadresse, rel. zum Strukturanfang = : gesamter Speicherbedarf einer Struktur	Systemeintrag: Die Adresse wird vom System vergeben und angezeigt, wenn Sie die Eingabe einer Deklaration beenden.
Variable	Symbolischer Name der Variablen	Der Name muß mit einem Buchstaben beginnen. Reservierte Schlüsselwörter sind nicht erlaubt.	erforderlich
Deklaration	Deklarationstyp, ”Verwendungs-zweck” der Variablen	Möglich sind je nach Bausteinart: Eingangsparameter ”in” Ausgangsparameter ”out” Durchgangsparameter ”in_out” Statische Variablen ”stat” Temporäre Variablen ”temp”	Systemvorgabe entsprechend Bausteinart
Datentyp	Datentyp der Variable (BOOL, INT, WORD, ARRAY usw.).	Die elementaren Datentypen können Sie über das Menü der rechten Maustaste auswählen.	erforderlich
Anfangswert	Anfangswert, wenn die Software nicht den Default-Wert übernehmen soll.	Muß mit Datentyp kompatibel sein. Der Anfangswert wird beim erstmaligen Speichern eines DBs als aktueller Wert für die Variable übernommen, falls Sie nicht explizit einen aktuellen Wert festlegen.	optional
Kommentar	Kommentar zur Dokumentation		optional

Vorbelegung

Nachdem Sie einen neuerstellten Codebaustein geöffnet haben, wird eine voreingestellte Variablendeklarationstabelle angezeigt. Diese listet nur die für die gewählte Bausteinart zulässigen Deklarationstypen (in, out, in_out, stat, temp) auf und zwar in der vorgeschriebenen Reihenfolge. Bei der Neuerstellung eines OBs wird eine Standard-Variablendeklaration angezeigt, deren Werte Sie ändern können.

Nicht editierbare Spalten in der Deklarationstabelle

Spalte	Eintrag
Adresse	Der Eintrag wird vom System vergeben und angezeigt, wenn Sie die Eingabe einer Deklaration beenden.
Deklarationstyp	Der Deklarationstyp wird durch die Position der Deklaration innerhalb der Tabelle bestimmt. Dadurch wird sichergestellt, daß Variablen nur in der richtigen Reihenfolge der Deklarationstypen eingegeben werden können. Wollen Sie den Deklarationstyp einer Deklaration ändern, schneiden Sie die Deklaration zunächst aus und fügen Sie diese danach unter dem neuen Deklarationstyp wieder ein.

9.2.4 Allgemeine Hinweise zu Variablendeklarationstabellen

Zur Bearbeitung der Tabelle stehen Ihnen die bekannten Funktionen des Menüs **Bearbeiten** zur Verfügung. Zur einfachen Bearbeitung können Sie auf das kontextsensitive Menü der rechten Maustaste zugreifen. Bei der Eingabe des Datentyps werden Sie ebenfalls durch das Menü der rechten Maustaste unterstützt.

Markieren in Variablendeklarationstabellen

Einzelne Zeilen markieren Sie, indem Sie auf das zugehörige, schreibgeschützte Adreßfeld klicken. Zusätzliche Zeilen desselben Deklarationstyps markieren Sie bei gedrückter SHIFT-Taste. Die markierten Zeilen werden schwarz hinterlegt.

ARRAYs markieren Sie durch Klick auf das Adreßfeld der betreffenden Zeile.

Wollen Sie eine **Struktur markieren**, dann führen Sie einen Mausklick auf das Adreßfeld der ersten oder letzten Zeile aus (in der also das Schlüsselwort STRUCT oder END_STRUCT steht). Einzelne Deklarationen innerhalb einer Struktur markieren Sie durch einen Mausklick auf das entsprechende Adreßfeld der Zeile.

Wenn Sie Strukturen innerhalb anderer Strukturen eingeben, wird Ihnen die Hierarchie durch entsprechende Einrückung der Variablennamen angezeigt.

Rückgängig machen von Aktionen

In der Variablendeklarationstabelle können Sie mit dem Menübefehl **Bearbeiten > Rückgängig** jeweils die letzte Ausschneide- oder Lösch-Operation rückgängig machen.

9.3 Multiinstanzen in der Variablendeklarationstabelle

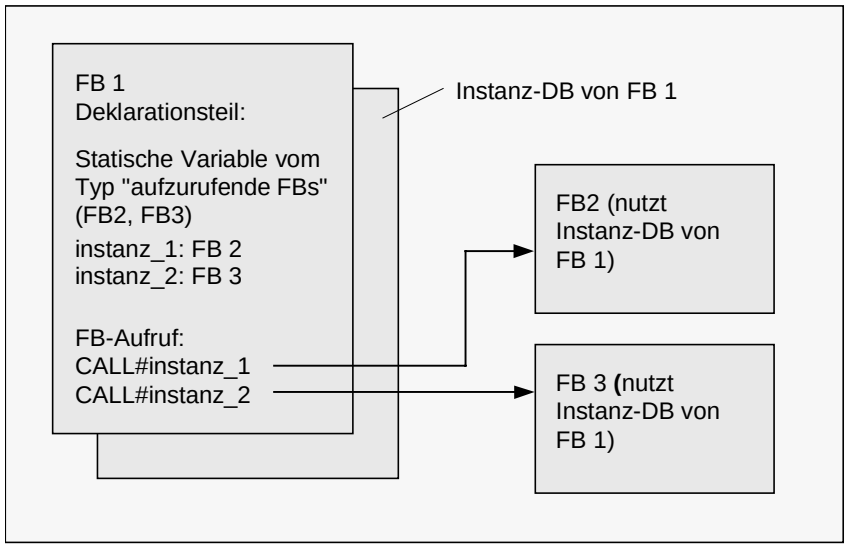
9.3.1 Verwendung von Multiinstanzen

Möglicherweise wollen oder können Sie aufgrund der Leistungsdaten (z.B. Speicherplatz) der verwendeten S7-CPUs nur eine beschränkte Anzahl von Datenbausteinen für Instanzdaten spendieren. Wenn in Ihrem Anwenderprogramm in einem FB weitere, bereits vorhandene Funktionsbausteine aufgerufen werden (Aufrufhierarchie von FBs), dann können Sie diese weiteren Funktionsbausteine ohne eigene (d. h. zusätzliche) Instanz-DBs aufrufen.

Dazu bietet sich folgende Lösung an:

- Nehmen Sie die aufzurufenden FBs als statische Variablen in die Variablendeklaration des aufrufenden FBs auf.
- In diesem Funktionsbaustein rufen Sie weitere Funktionsbausteine auf ohne eigene (d. h. zusätzliche) Instanz-DBs.
- Damit erreichen Sie eine Konzentrierung der Instanzdaten in einem Instanz-Datenbaustein, d.h. Sie können die verfügbare Anzahl der DBs besser ausnutzen.

Das folgende Beispiel zeigt die beschriebene Lösung: FB 2 und FB 3 nutzen den Instanz-DB des Funktionsbausteins FB 1, von dem aus sie aufgerufen wurden.



Einzige Voraussetzung: Sie müssen dem aufrufenden Funktionsbaustein "mitteilen", welche Instanzen Sie aufrufen und von welchem (FB-) Typ diese Instanzen sind. Die Angaben sind im Deklarationsteil des aufrufenden FBs zu machen. Der zu verwendende FB muß mindestens eine Variable oder einen Parameter aus dem Datenbereich (also nicht VAR_TEMP) haben.

Verwenden Sie keine Multiinstanzdatenbausteine, solange Online-Änderungen bei laufender CPU zu erwarten sind. Ein stoßfreies Nachladen ist nur bei Verwendung von Instanzdatenbausteinen sichergestellt.

9.3.2 Regeln für die Bildung von Multiinstanzen

Für die Deklaration von Multiinstanzen gelten folgende Regeln:

- Die Deklaration von Multiinstanzen ist nur in Funktionsbausteinen möglich, die mit STEP 7 ab Version 2 erstellt wurden (siehe Bausteinattribut in den Eigenschaften des FBs).
- Für das Deklarieren von Multiinstanzen muß der Funktionsbaustein als multiinstanzfähiger FB erzeugt worden sein (Voreinstellung ab STEP 7 Version x.x; abschaltbar im Editor über **Extras > Einstellungen**).
- Dem Funktionsbaustein, in dem eine Multiinstanz deklariert wird, muß ein Instanz-DB zugeordnet werden.
- Eine Multiinstanz kann nur als statische Variable deklariert werden (Deklarationstyp ”stat”).

Hinweis

- Sie können auch Multiinstanzen zu Systemfunktionsbausteinen erzeugen.
- Wurde der FB als nicht multiinstanzfähig angelegt und Sie benötigen nun doch diese Eigenschaft, so können Sie aus dem FB eine Quelle generieren, in dieser die Bausteineigenschaft CODE_VERSION1 löschen und daraus wieder den FB übersetzen.

9.3.3 Eingeben der Multiinstanz in der Variablendeklarationstabelle

1. Öffnen Sie den FB, von dem aus die unterlagerten FBs aufgerufen werden sollen.
2. Legen Sie in der Variablendeklaration des aufrufenden FBs eine statische Variable für jeden Aufruf eines Funktionsbausteins fest, für dessen Instanz Sie keinen Instanz-Datenbaustein angeben wollen:
   - Positionieren Sie in eine Leerzeile mit der Deklaration "stat" in der 2. Spalte.
   - Geben Sie hinter dem Deklarationstyp ”stat” in Spalte "Name” eine Bezeichnung für den FB-Aufruf ein.
   - Geben Sie in Spalte “Typ” den aufzurufenden FB absolut oder mit seinem symbolischen Namen an.
   - Eventuelle Erläuterungen können Sie in die Kommentarspalte eintragen.

Aufrufe im Anweisungsteil

Wenn Sie Multiinstanzen deklariert haben, können Sie FB-Aufrufe ohne Angabe eines Instanz-DB verwenden.

Beispiel: Ist die statische Variable "Name: Motor_1 , Datentyp: FB20" definiert, läßt sich die Instanz wie folgt aufrufen:

```
Call Motor_1        // Aufruf des FB 20 ohne Instanz-DB
```

9.4 Allgemeine Hinweise zum Editieren von Anweisungen und Kommentaren

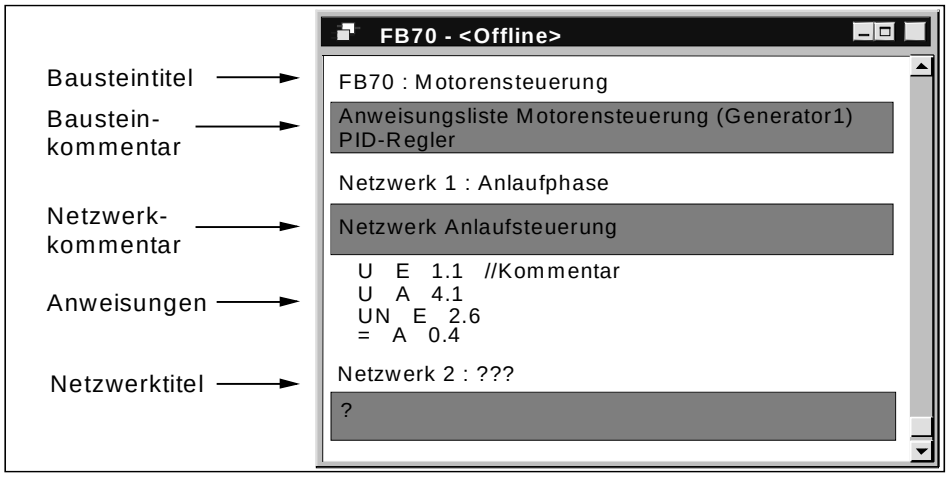
9.4.1 Aufbau des Anweisungsteils

Im Anweisungsteil beschreiben Sie den Programmablauf Ihres Codebausteins. Sie geben dazu die jeweiligen Anweisungen in Abhängigkeit der gewählten Programmiersprache in Netzwerken ein. Der Editor führt nach der Eingabe einer Anweisung sofort eine Syntaxprüfung durch und zeigt Ihnen Fehler rot und kursiv an.

Der Anweisungsteil für einen Codebaustein besteht in den häufigsten Fällen aus mehreren Netzwerken, die wiederum aus einer Liste von Anweisungen bestehen.

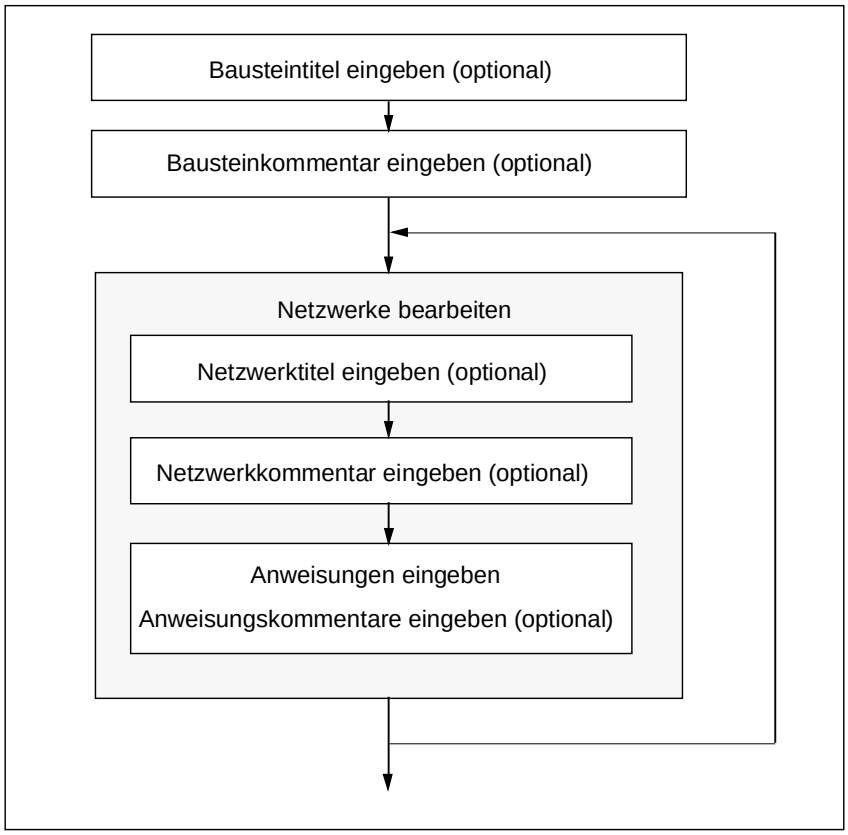
In einem Anweisungsteil können Sie Bausteintitel, Bausteinkommentare, Netzwerktitel, Netzwerkkommentare und Anweisungszeilen innerhalb der Netzwerke bearbeiten.

Aufbau des Anweisungsteils am Beispiel der Programmiersprache AWL



9.4.2 Vorgehensweise beim Eingeben von Anweisungen

Die Bestandteile des Anweisungsteils können Sie grundsätzlich in beliebiger Reihenfolge bearbeiten. Wenn Sie zum ersten Mal einen Baustein programmieren, empfiehlt sich folgendes Vorgehen:



Änderungen können Sie zusätzlich zum Einfügemodus auch im Überschreibmodus vornehmen. Zwischen beiden Modi wechseln Sie mit der INSERT-Taste.

9.4.3 Eingeben von globalen Symbolen in ein Programm

Mit dem Menübefehl **Einfügen > Symbol** können Sie Symbole in den Anweisungsteil Ihres Programms einfügen. Steht der Cursor am Beginn, am Ende oder innerhalb einer Zeichenkette, so ist bereits das Symbol markiert, das mit dieser Zeichenkette beginnt - falls ein solches Symbol existiert. Wenn Sie die Zeichenkette ändern, wird die Markierung in der Liste nachgeführt.

Trennzeichen für Beginn und Ende einer Zeichenkette sind z.B. Leerzeichen, Punkt, Doppelpunkt. Innerhalb von globalen Symbolen werden keine Trennzeichen interpretiert.

Beim Einfügen von Symbolen können Sie wie folgt vorzugehen:

1. Geben Sie die Anfangsbuchstaben des gewünschten Symbols im Programm ein.
2. Drücken Sie gleichzeitig die CTRL-Taste und die J-Taste, um die Liste der Symbole anzuzeigen. Das erste Symbol mit den eingegebenen Anfangsbuchstaben ist bereits markiert.
3. Übernehmen Sie das Symbol durch Drücken der RETURN-Taste oder wählen Sie ein anderes Symbol aus.

Daraufhin wird das in doppelte Hochkommas eingeschlossene Symbol anstelle der Anfangsbuchstaben eingefügt.

Generell gilt: Steht der Cursor am Beginn, am Ende oder innerhalb einer Zeichenkette, so wird beim Einfügen eines Symbols diese Zeichenkette durch das in doppelte Hochkommas eingeschlossene Symbol ersetzt.

9.4.4 Titel und Kommentare zu Bausteinen und Netzwerken

Kommentare erhöhen die Lesbarkeit Ihres Anwenderprogramms und ermöglichen somit eine effektivere Inbetriebnahme und evt. Fehlersuche. Sie sind ein wichtiger Bestandteil der Programm-Dokumentation und sollten auf jeden Fall verwendet werden.

Kommentare für KOP, FUP und AWL-Programme

Folgende Kommentare stehen Ihnen zur Verfügung:

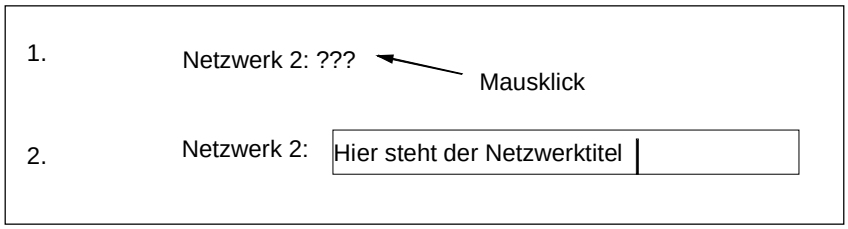
- Bausteintitel: Überschrift eines Bausteins (max.64 Zeichen).
- Bausteinkommentar: Dokumentieren des gesamten Codebausteins, wie z. B. der Zweck des Bausteins.
- Netzwerktitel: Überschrift eines Netzwerks (max. 64 Zeichen).
- Netzwerkkommentar: Dokumentieren der Funktion einzelner Netzwerke.
- Kommentarspalte der Variablendeklarationstabelle: Kommentare für die deklarierten Lokaldaten.
- Symbolkommentar: Kommentare, die bei der Definition des Namens in der Symboltabelle für einen Operanden angegeben worden sind.
  Sie können sich diese Kommentare mit dem Menübefehl **Ansicht > Symbolinformation** anzeigen lassen.

Bausteintitel und Netzwerktitel sowie Bausteinkommentare oder Netzwerkkommentare können Sie im Anweisungsteil eines Codebausteins eingeben.

Baustein- oder Netzwerktitel

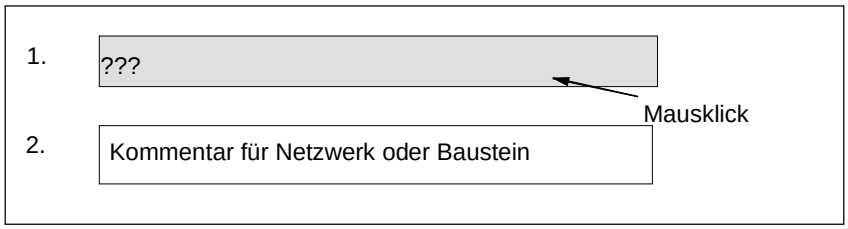
Um Baustein- oder Netzwerktitel einzugeben, positionieren Sie den Cursor auf die drei Fragezeichen rechts neben dem Bausteinnamen bzw. Netzwerknamen (z. B. Netzwerk 1 : ???). Es öffnet sich ein Textfeld, in das Sie den Titel eingeben. Dieser kann maximal 64 Zeichen lang sein.

Bausteinkommentare beziehen sich auf den gesamten Codebaustein. Dort können Sie die Funktion desselben beschreiben. Netzwerkkommentare beziehen sich auf einzelne Netzwerke und beschreiben deren Einzelheiten.



-Baustein- und Netzwerkkommentare

Sie können die Ansicht der grauen Kommentarfelder über den Menübefehl **Ansicht > Anzeigen mit > Kommentar** ein- bzw. ausblenden. Durch einen Doppelklick auf ein solches Kommentarfeld wird das Textfeld geöffnet, in das Sie nun Ihre Erläuterungen eingeben können. Pro Baustein stehen Ihnen 64 KByte für Baustein- und Netzwerkkommentare zur Verfügung.



9.4.5 Suchfunktion für Fehler im Anweisungsteil

Aufgrund der roten Darstellung sind Fehler im Anweisungsteil leicht zu erkennen. Um leichter zu Fehlern zu navigieren, die außerhalb des gerade sichtbaren Bereiches liegen, bietet der Editor die zwei Suchfunktionen **Bearbeiten > Gehe zu > Vorheriger Fehler/Nächster Fehler** an.

Die Fehlersuche ist netzwerkübergreifend. Das heißt, es wird im gesamten Anweisungsteil gesucht, nicht nur innerhalb eines Netzwerkes oder des gerade sichtbaren Bereichs.

Wenn Sie die Statuszeile mit dem Menübefehl **Ansicht > Statuszeile** einschalten, werden Ihnen dort Hinweise zu den Fehlern angezeigt.

Fehlerkorrekturen und Änderungen können Sie auch im Überschreibmodus vornehmen. Zwischen Einfüge- und Überschreibmodus wechseln Sie mit der INSERT-Taste.

9.5 Editieren von KOP-Anweisungen im Anweisungsteil

9.5.1 Einstellungen für Programmiersprache KOP

Einstellen des Layout für KOP

Sie können für die Programmerstellung in der Programmiersprache KOP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat / Querformat / maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren KOP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen**.
2. Wählen Sie im nachfolgenden Dialogfeld das Register "KOP/FUP".
3. Wählen Sie im Listenfeld Layout das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Einstellungen zum Drucken:

Wenn Sie den KOP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Einstellungen im Register "KOP/FUP"

Im Register ”KOP/FUP”, auf das Sie über **Extras > Einstellungen** zugreifen, können Sie Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

9.5.2 Regeln für die Eingabe von KOP-Anweisungen

Die Beschreibung der Sprache "KOP" finden Sie im Handbuch "KOP für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu KOP.

Ein KOP-Netzwerk kann aus mehreren Elementen in mehreren Zweigen bestehen. Alle Elemente und Zweige müssen untereinander verbunden sein, wobei die linke Stromschiene nicht als Verbindung zählt (IEC 1131-3).

Bei der Programmierung in KOP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch Fehlermeldung hingewiesen.

Abschluß eines KOP-Netzwerks

Jedes KOP-Netzwerk muß einen Abschluß in Form einer Spule oder einer Box haben. Folgende KOP-Elemente dürfen Sie jedoch nicht als Abschluß eines Netzwerkes verwenden:

- Vergleicherboxen
- Spulen für Konnektoren _/(#)_/
- Spulen für positive _/(P)_/ oder negative _/(N)_/ Flankenauswertung

Plazierung von Boxen

Der Ausgangspunkt des Zweiges für einen Boxanschluß muß immer die linke Stromschiene sein. Im Zweig vor der Box dürfen jedoch logische Verknüpfungen oder andere Boxen vorhanden sein.

Plazierung von Spulen

Spulen werden automatisch an den rechten Rand des Netzwerks plaziert, wo sie den Abschluß eines Zweiges bilden.

Ausnahmen: Spulen für Konnektoren _/(#)_/ und positive _/(P)_/ oder negative _/(N)_/ Flankenauswertung können weder ganz links noch ganz rechts im Zweig plaziert werden. Auch in Parallelzweigen sind sie nicht zulässig.

Unter den Spulen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

- Spulen mit erforderlicher boolescher Verknüpfung:
  - Ausgang _/( ), Ausgang setzen _/(S), Ausgang rücksetzen _/(R)
  - Konnektor _/(#)_/, positive Flanke _/(P)_/, negative Flanke _/(N)_/
  - alle Zähler- und Timerspulen
  - Springe wenn 0 _/(JMPN)
  - Master Control Relais einschalten _/(MCR<)
  - Verknüpfungsergebnisses ins BR-Register laden _/(SAVE)
  - Springe zurück _/(RET)

- Spulen, die keine boolesche Verknüpfung erlauben:
  - Master Control Relais Anfang _/(MCRA)
  - Master Control Relais Ende _/(MCRD)
  - Datenblock Öffnen _/(OPN)
  - Master Control Relais ausschalten _/(MCR>)

Alle anderen Spulen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Folgende Spulen dürfen Sie **nicht als Parallelausgang** verwenden:

- Springe wenn 0 _/(JMPN)
- Springe wenn 1 _/(JMP)
- Bausteinaufruf _/(CALL)
- Rücksprung _/(RET)

Freigabeeingang/ Freigabeausgang

Die Beschaltung des Freigabeeingangs ”EN” bzw. des Freigabeausgangs ”ENO” von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Besteht eine Verzweigung aus nur einem Element, so wird mit dem Entfernen dieses Elementes der gesamte Zweig entfernt.

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

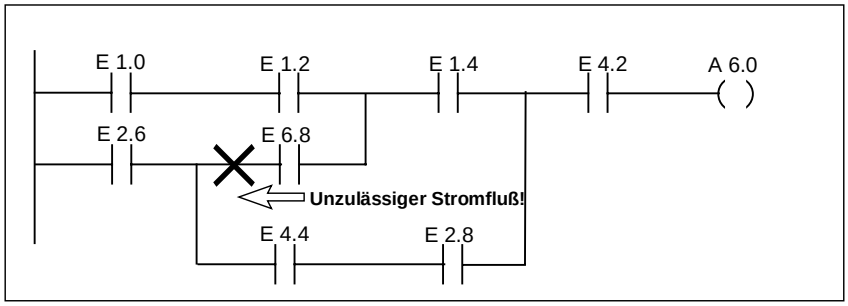
Parallele Verzweigungen

- Ergänzen Sie ODER-Verzweigungen von links nach rechts.
- Parallelzweige werden nach unten geöffnet und nach oben geschlossen.
- Parallelzweige werden immer hinter dem markierten KOP-Element geöffnet.
- Parallelzweige werden immer hinter dem markierten KOP-Element geschlossen.
- Um einen Parallelzweig zu löschen, müssen Sie alle KOP-Elemente dieses Zweiges löschen. Beim Entfernen des letzten KOP-Elements der Verzweigung wird der Rest der Verzweigung ebenfalls entfernt.

9.5.3 Unzulässige Verschaltungen in KOP

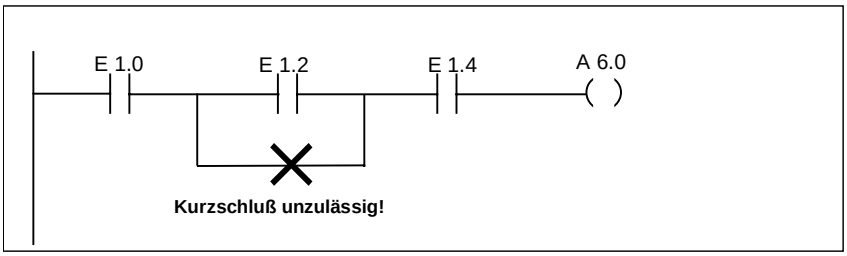
Stromfluß von rechts nach links

Es können keine Zweige editiert werden, die einen Stromfluß in umgekehrter Richtung verursachen könnten. Bild 3-8 zeigt ein Beispiel. Beim Signalzustand ”0” an E 1.4 käme es an E 6.8 zu einem Stromfluß von rechts nach links. Dieses ist nicht erlaubt.



Kurzschluß

Es können keine Zweige editiert werden, die einen Kurzschluß verursachen. Nachfolgendes Bild zeigt ein Beispiel:



9.6 Editieren von FUP-Anweisungen im Anweisungsteil

9.6.1 Einstellungen für die Programmiersprache FUP

Einstellen des Layout für FUP

Sie können für die Programmerstellung in der Programmiersprache FUP das Layout festlegen. Das ausgewählte Format (DIN A4 Hochformat/Querformat/maximale Größe) wirkt sich auf die Anzahl der in einem Zweig darstellbaren FUP-Elemente aus.

1. Wählen Sie den Menübefehl **Extras > Einstellungen**.
2. Wählen Sie im nachfolgenden Dialogfeld das Register "KOP/FUP".
3. Wählen Sie im Listenfeld Layout das gewünschte Format. Geben Sie die gewünschte Formatgröße an.

Hinweise zum Drucken

Wenn Sie den FUP-Anweisungsteil ausdrucken möchten, so sollten Sie bereits vor der Erstellung des Anweisungsteils das geeignete Seitenformat einstellen.

Register KOP/FUP unter Extras > Einstellungen

Im Register ”KOP/FUP”, auf das Sie über **Extras > Einstellungen** zugreifen, können Sie Grundeinstellungen vornehmen, z. B. zum Layout und zur Operandenfeldbreite.

9.6.2 -Regeln für die Eingabe von FUP-Anweisungen

Die Beschreibung der Sprache "FUP" finden Sie im Handbuch "FUP für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu FUP.

Ein FUP-Netzwerk kann aus mehreren Elementen bestehen. Alle Elemente müssen untereinander verbunden sein (IEC 1131-3).

Bei der Programmierung in FUP müssen Sie einige Regeln beachten. Auf eventuelle Fehler werden Sie durch Fehlermeldung hingewiesen.

Einfügen und Editieren von Adressen und Parametern

Beim Einfügen eines FUP-Elements werden die Zeichenfolgen "???" und "..." als Platzhalter für Adressen bzw. Parameter eingesetzt.

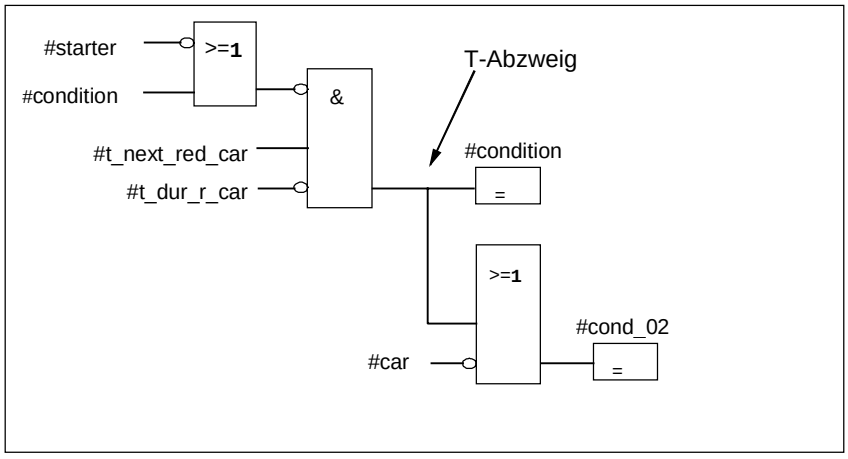
- Die rot dargestellte Zeichenfolge "???" kennzeichnet Adressen und Parameter, die beschaltet werden müssen.
- Die schwarz dargestellte Zeichenfolge "..." kennzeichnet Adressen und Parameter, die beschaltet werden können.

Wenn Sie den Mauszeiger über den Platzhalter bewegen, wird der erwartete Datentyp angezeigt.

Plazierung von Boxen

An Boxen mit binären Verknüpfungen (&, >=1, XOR) können Standard-Boxen (Flipflops, Zähler, Zeiten, Rechenoperationen usw.) angefügt werden. Ausgenommen von dieser Regelung sind Vergleicherboxen.

In einem Netzwerk dürfen keine voneinander getrennten Verknüpfungen mit separaten Ausgängen programmiert werden. Sie können jedoch mit Hilfe des T-Abzweigs einer Verknüpfungskette mehrere Zuweisungen zuordnen. Nachfolgendes Bild zeigt ein Netzwerk mit zwei Zuweisungen.



Die folgenden Boxen dürfen nur am rechten Rand der Verknüpfungskette plaziert werden, wo sie den Abschluß der Kette bilden:

- Zähleranfangswert setzen
- Vorwärtszählen, Rückwärtszählen
- Zeit als Impuls starten, Zeit als verlängerten Impuls starten
- Zeit als Einschaltverzögerung/Ausschaltverzögerung starten.

Unter den Boxen gibt es solche, die eine boolesche Verknüpfung verlangen, und solche, die keine boolesche Verknüpfung haben dürfen.

Boxen mit erforderlicher boolescher Verknüpfung:

- Ausgang, Ausgang setzen, Ausgang rücksetzen _/[R]
- Konnektor _/[#]_/, positive Flanke _/[P]_/, negative Flanke _/[N]_/
- alle Zähler- und Timerboxen
- Springe wenn 0 _/[JMPN]
- Master Control Relais einschalten _/[MCR<]
- Verknüpfungsergebnis ins BIE-Register laden _/[SAVE]
- Springe zurück _/[RET]

Boxen, die keine boolesche Verknüpfung erlauben:

- Master Control Relais Anfang [MCRA]
- Master Control Relais Ende [MCRD]
- Datenblock öffnen [OPN]
- Master Control Relais ausschalten [MCR>]

Alle anderen Boxen können wahlweise eine boolesche Verknüpfung haben oder auch nicht.

Freigabeeingang/-Freigabeausgang

Die Beschaltung des Freigabeeingangs ”EN” bzw. des Freigabeausgangs ”ENO” von Boxen kann erfolgen, ist aber nicht zwingend erforderlich.

Entfernen und Ändern

Mit dem Entfernen einer Box werden mit Ausnahme des Hauptzweiges auch alle Zweige, die mit booleschen Eingängen der Box verbunden sind, entfernt.

Zum einfachen Austauschen von Elementen gleichen Typs steht Ihnen der Überschreibmodus zur Verfügung.

9.7 Editieren von AWL-Anweisungen im Anweisungsteil

9.7.1 Einstellungen für die Programmiersprache AWL

Einstellen der Mnemonik

Sie können zwischen zwei Einstellungen der Mnemonik wählen:

- Deutsch oder
- Englisch.

Die Mnemonik stellen Sie vor dem Öffnen eines Bausteins im SIMATIC Manager mit dem Menübefehl **Extras > Einstellungen** im Register ”Sprache” ein. Während der Bausteinbearbeitung kann die Mnemonik nicht geändert werden.

Die **Bausteineigenschaften** bearbeiten Sie über einen eigenen Dialog.

Im Editor können Sie mehrere Bausteine geöffnet haben und nach Wunsch abwechselnd bearbeiten.

9.7.2 Regeln für die Eingabe von AWL-Anweisungen

Die Beschreibung der Sprache "AWL" finden Sie im Handbuch "AWL für S7-300/400 - Bausteine programmieren" oder in der Online-Hilfe zu AWL (Sprachbeschreibungen)

Bei der inkrementellen Eingabe von AWL-Anweisungen müssen Sie folgende Grundregeln beachten: .

- Achten Sie auf die Reihenfolge der Programmierung der Bausteine. Aufgerufene Bausteine müssen vor den aufrufenden Bausteinen programmiert werden.
- Eine Anweisung besteht aus der Angabe von Sprungmarke (optional), Operation, Operand und Kommentar (optional).
  Beispiel: M001: U E1.0 //Kommentar
- Jede Anweisung steht in einer eigenen Zeile.
- Sie können je Baustein maximal 999 Netzwerke eingeben.
- Je Netzwerk können Sie etwa 2000 Zeilen eingeben. Bei vergrößerter oder verkleinerter Ansicht sind entsprechend mehr oder weniger Zeilen möglich.
- Bei der Eingabe von Operationen oder absoluten Adressen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

9.8 Aktualisieren von Bausteinaufrufen

9.8.1 Aktualisieren von Bausteinaufrufen

Mit dem Menübefehl **Bearbeiten > Aufruf > Aktualisieren** in "KOP/AWL/FUP-Bausteine programmieren" können Sie ungültig gewordene Bausteinaufrufe oder UDT-Verwendungen nach folgenden Schnittstellenänderungen automatisch abgleichen:

- Einfügen neuer Parameter
- Löschen von Parametern
- Namensänderung von Parametern
- Typänderung von Parameters
- Änderung der Reihenfoge (Umkopieren) von Parametern

Bei der Zuordnung von Formal- und Aktualseite wird nach folgenden Regeln in der angegebenen Reihenfolge vorgegangen:

1. **Gleiche Parameternamen:**
   Die Aktualparameter werden automatisch zugeordnet, wenn der Name des Formalparameters gleich geblieben ist.
   Sonderfall: Die Vorverknüpfung von binären Eingangsparametern kann in KOP und FUP nur bei gleichem Datentyp (BOOL) automatisch zugeordnet werden. Wurde der Datentyp in einem solchen Fall geändert, so bleibt die Vorverknüpfung als offener Zweig bestehen.
2. **Gleiche Parameterdatentypen:**
   Nachdem die Parameter mit gleichen Namen zugeordnet wurden werden nun noch nicht zugeordnete Aktualparameter zu Formalparametern mit gleichem Datentyp wie der ”alte” Formalparameter zugeordnet.
3. **Gleiche Parameterposition:**
   Die nach der Zuordnung gemäß Regel 1 und 2 noch nicht zugeordnete Aktualparameter werden nun gemäß ihrer Parameterposition in der ”alten” Schnittstelle den neuen Formalseiten zugeordnet.
4. Können Aktualparameter nach den drei genannten Regeln nicht zugeordnet werden, so werden diese gelöscht bzw. bleiben im Falle von binären Vorverknüpfungen in KOP oder FUP als offene Zweige erhalten.

Überprüfen Sie nach Ausführung dieser Funktion die vorgenommenen Änderungen in der Variablendeklarationstabelle und im Anweisungsteil des Programms.

9.9 Speichern von Codebausteinen

9.9.1 Speichern von Codebausteinen

Um neu erstellte Bausteine bzw. Änderungen im Anweisungsteil von Codebausteinen oder in Deklarationstabellen in die PG-Datenhaltung zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Dazu werden die Daten auf die Festplatte des PGs geschrieben.

Speichern von Bausteinen auf die Festplatte des PGs:

1. Aktivieren Sie das Arbeitsfenster des zu speichernden Bausteins.
2. Wählen Sie:
   - den Menübefehl **Datei > Speichern**, wenn Sie den Baustein unter gleichem Namen abspeichern möchten.
   - den Menübefehl **Datei > Speichern** unter, wenn Sie den Baustein unter einem anderen S7-Anwenderprogramm bzw. unter anderem Namen abspeichern möchten. Geben Sie im folgenden Dialogfeld den neuen Pfad bzw. den neuen Baustein an.

In beiden Fällen wird der Baustein nur bei fehlerloser Syntax gespeichert. Syntaktische Fehler werden bei der Erstellung sofort erkannt und rot dargestellt. Diese Fehler müssen vor dem Speichern des Bausteins behoben sein.

Hinweis

- Das Speichern von Bausteinen oder Quellen unter anderen Projekten oder Bibliotheken können Sie auch im SIMATIC Manager durchführen (z.B. durch Drag&Drop).
- Das Speichern von Bausteinen oder kompletten Anwenderprogrammen auf eine Memory Card wird nur im SIMATIC Manager durchgeführt.
- Wenn beim Speichern oder Übersetzen großer Bausteine Probleme auftreten, sollten Sie das Projekt reorganisieren. Rufen Sie dazu den Menübefehl **Datei > Reorganisieren** im SIMATIC Manager auf. Versuchen Sie anschließend das Speichern bzw. Übersetzen erneut.

9.9.2 Korrigieren der Schnittstellen in einem FC, FB oder UDT

Wenn Sie die Schnittstelle in einem FB, FC oder UDT korrigieren müssen, so gehen Sie wie folgt vor, um Zeitstempelkonflikte zu vermeiden:

1. Generieren Sie eine AWL-Quelle mit dem zu ändernden Baustein und allen direkt oder indirekt referenzierenden Bausteinen.
2. Speichern Sie die Änderungen in der erzeugten Quelle.
3. Übersetzen Sie die angepaßte Quelle zurück in die Bausteine.

Nun können Sie die Schnittstellenänderung speichern/laden.

9.9.3 Vermeiden von Fehlern beim Aufrufen von Bausteinen

STEP 7 überschreibt Daten in DB-Register

STEP 7 ändert die Register der S7-300/S7-400-CPU bei verschiedenen Operationen. Die Inhalte der DB- und DI-Register beispielsweise werden getauscht, wenn Sie einen FB aufrufen. Dadurch kann der Instanz-DB des aufgerufenen FB geöffnet werden, ohne die Adresse des vorherigen Instanz-DB zu verlieren.

Wenn Sie mit der absoluten Adressierung arbeiten, können Fehler beim Zugreifen auf Daten auftreten, die in den Registern gespeichert sind: in einigen Fällen werden die Adressen in dem Register AR1 (Adreßregister 1) und in dem DB-Register überschrieben. Dadurch kann es sein, daß Sie falsche Adressen lesen oder in falsche Adressen schreiben.

Gefahr

Gefahr von Sachschäden und Personenschäden bei der Verwendung von:

1. CALL FC, CALL FB, CALL Multiinstanz
2. vollqualifizierte DB-Zugriffe (z. B. DB20.DBW10)
3. Zugriffe auf Variablen eines zusammengesetzten Datentyps

Es kann dabei geschehen, daß die Inhalte von DB-Register (DB und DI), Adreßregister (AR1, AR2) und Akkus (AKKU1, AKKU2) verändert werden.

Ebenso kann beim FB-CALL/FC-CALL das Verknüpfungsergebnis VKE nicht als zusätzlicher (impliziter) Parameter verwendet werden.

Wenn Sie die oben genannten Programmiermöglichkeiten nutzen, müssen Sie selbst für eine Wiederherstellung der Inhalte Sorge tragen, da es sonst zu einem Fehlverhalten kommen kann.

Speichern von korrekten Daten

Kritisch wird es mit den Inhalten des DB-Registers, wenn Sie im abgekürzten Format der absoluten Adressen auf Daten zugreifen. Gehen Sie beispielsweise davon aus, daß DB20 geöffnet ist (und seine Nummer im DB-Register gespeichert ist), können Sie DBX0.2 angeben, um auf die Daten zuzugreifen, die in Bit 2 von Byte 0 des DB gespeichert sind, dessen Adresse in dem DB-Register abgelegt ist (also DB20). Enthält das DB-Register allerdings eine andere DB-Nummer, greifen Sie auf die falschen Daten zu.

Fehler beim Zugreifen auf Daten des DB-Registers können Sie vermeiden, wenn Sie die folgende Methode zum Adressieren von Daten verwenden:

- Verwenden Sie symbolische Adressierung
- Verwenden Sie die vollständige absolute Adresse (z. B. *DB20.DBX0.2*)

Bei diesen beiden Adressierungsmethoden öffnet STEP 7 automatisch den richtigen DB. Wenn Sie das Register AR1 für die indirekte Adressierung verwenden, müssen Sie immer die richtige Adresse in AR1 laden.

Situationen, in denen Register modifiziert werden

Die Manipulation der Adreßregister zur indirekten Adressierung ist ausschließlich in AWL relevant. Die anderen Sprachen unterstützen keinen indirekten Zugriff auf die Adreßregister.

Die Anpassung des DB-Registers durch den Compiler muß bei allen Programmiersprachen beachtet werden, um eine korrekte Parameterübergabe bei Bausteinaufrufen sicherzustellen.

In den folgenden Situationen werden die Inhalte des Adreßregisters AR1 und des DB-Registers des aufrufenden Bausteins überschrieben:

Situation	Erläuterung
Bei Aktualparametern aus einem DB	• Nachdem Sie einem Baustein einen Aktualparameter zugeordnet haben, der in einem DB gespeichert ist (z.B. DB20.DBX0.2), öffnet STEP 7 diesen DB (DB20) und paßt dabei den Inhalt des DB-Registers an. Das Programm arbeitet im Anschluß an den Bausteinaufruf dann mit dem angepaßten DB.
Bei Aufruf von Bausteinen in Zusammenhang mit höheren Datentypen	• Nach einem Bausteinaufruf aus einem FC, der eine Komponente eines Formalparameters eines höheren Datentyps (String, Array, Struct oder UDT) an den aufgerufenen Baustein übergibt, wird der Inhalt von AR1 und des DB Registers des aufrufenden Bausteins modifiziert. • Dasselbe gilt bei Aufruf aus einem FB, wenn der Parameter im VAR_IN_OUT Bereich des Aufrufers liegt.
Bei Zugriff auf Komponenten höheren Datentyps	• Beim Zugriff eines FB auf eine Komponente eines Formalparameters höheren Datentyps im VAR_IN_OUT Bereich (String, Array, Struct oder UDT) verwendet STEP7 das Adreßregister AR1 und das DB Register. Dadurch werden die Inhalte der beiden Register modifiziert. • Beim Zugriff eines FC auf eine Komponente eines Formalparameters höheren Datentyps (String, Array, Struct oder UDT) verwendet STEP7 das Adreßregister AR1 und das DB Register. Dadurch werden die Inhalte der beiden Register modifiziert.

Hinweis

- Beim Aufruf eines FB aus einem Baustein mit Bausteinversion 1 wird der Aktualparameter für den ersten booleschen IN bzw. IN_OUT Parameter nicht korrekt übergeben, wenn der Befehl vor dem Call nicht VKE begrenzend ist. In diesem Fall wird er mit dem anstehenden VKE verknüpft.
- Beim Aufruf eines FB (Single oder Multiinstanz) wird das Adreßregister AR2 beschrieben.
- Wird innerhalb eines FB das Adreßregister AR2 modifiziert, ist die ordnungsgemäße Bearbeitung dieses FB nicht mehr gewährleistet.
- Wird einem ANY-Parameter nicht die vollständige absolute DB-Adresse übergeben, enthält der ANY-Pointer nicht die DB-Nummer des aufgeschlagenen DBs, sondern immer die Nummer 0.

10 Erstellen von Datenbausteinen

10.1 Grundlagen zum Erstellen von Datenbausteinen

Der Datenbaustein ist ein Baustein, indem Sie z.B. Werte hinterlegen, auf die Ihre Maschine oder Anlage zurückgreift. Im Gegensatz zum Codebaustein, der mit einer der Programmiersprachen KOP/FUP oder AWL programmiert wurde, enthält der Datenbaustein nur den Variablendeklarationsteil. Der Anweisungsteil entfällt und damit auch das Programmieren von Netzwerken.

Nach dem Öffnen eines Datenbausteins können Sie sich den Baustein entweder in der Deklarationssicht oder in der Datensicht anzeigen lassen. Umschalten können Sie mit den Menübefehlen **Ansicht > Deklarationssicht** und **Ansicht > Datensicht**.

Deklarationssicht

Wählen Sie die Deklarationssicht, um

- die Datenstruktur globaler DBs abzulesen oder festzulegen,
- die Datenstruktur von DBs mit zugeordnetem anwenderdefinierten Datentyp (UDT) abzulesen,
- die Datenstruktur von DBs mit zugeordnetem Funktionsbaustein abzulesen.

Die Struktur von Datenbausteinen, die einem FB oder einem anwenderdefiniertem Datentyp zugeordnet sind, kann nicht geändert werden. Dazu müßte zunächst der entsprechende FB bzw. UDT geändert und anschließend der DB neu angelegt werden.

Datensicht

Wählen Sie die Datensicht, um Daten zu ändern. Nur in der Datensicht können Sie sich für jedes Element den aktuellen Wert anzeigen lassen, bzw. ihn eingeben oder ändern. In der Datensicht von Datenbausteinen werden bei Variablen mit zusammengesetzten Datentypen die Elemente einzeln mit vollständigem Namen aufgelistet.

Unterschied zwischen Instanz-Datenbaustein und globalen Datenbaustein

Der globale Datenbaustein ist keinem Codebaustein zugeordnet. Er enthält Werte, die von der Anlagen oder Maschine benötigt werden und läßt sich direkt an jeder Stelle des Programms aufrufen.

Der Instanz-Datenbaustein ist ein Baustein, der einem Codebaustein, z.B. einem Funktionsbaustein, unmittelbar zugeordnet ist. Der Instanz-Datenbaustein enthält die Daten, die in einem Funktionsbaustein in der Variablendeklarationstabelle hinterlegt wurden.

10.2 Deklarationssicht von Datenbausteinen

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Deklaration	Diese Spalte wird nur für Instanz-Datenbausteine dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: • Eingangsparameter (IN) • Ausgangsparameter (OUT) • Durchgangsparameter (IN_OUT) • statische Daten (STAT)
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangs-wert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 80 Zeichen lang sein.

10.3 Datensicht von Datenbausteinen

Die Datensicht zeigt Ihnen die aktuellen Werte aller Variablen des Datenbausteins. Sie können diese Werte nur in der Datensicht ändern. Die Tabellendarstellung dieser Sicht ist für alle Global-Datenbausteine gleich. Für Instanz-Datenbausteine wird zusätzlich die Spalte "Deklaration" angezeigt.

Für Variablen mit zusammengesetzten Datentypen oder anwenderdefinierten Datentypen werden in der Datensicht alle Elemente einzeln in einer eigenen Zeile mit vollständigem Namen dargestellt. Wenn die Elemente im IN_OUT-Bereich eines Instanz-Datenbausteins liegen, wird in der Spalte "Aktualwert" der Zeiger auf den zusammengesetzten oder anwenderdefinierten Datentyp angezeigt.

Die Datensicht zeigt folgende Spalten:

Spalte	**Erläuterung**
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Deklaration	Diese Spalte wird nur für Instanz-DBs dargestellt. Dieser Spalte können Sie entnehmen, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: • Eingangsparameter(IN) • Ausgangsparameter (OUT) • Durchgangsparameter (IN_OUT) • statische Daten (STAT)
Name	Hierbei handelt es sich um den Namen, der für die Variable festgelegt ist. Dieses Feld können Sie in der Datensicht nicht bearbeiten.
Typ	Hierbei handelt es sich um den Datentyp, der für die Variable festgelegt ist. Bei einem globalen Datenbaustein stehen hier nur noch elementare Datentypen, da in der Datensicht für Variablen mit zusammengesetzten oder anwenderdefinierten Datentypen die Elemente einzeln aufgelistet werden. Bei einem Instanz-Datenbaustein werden zusätzlich Parameterdatentypen angezeigt, bei Durchgangsparametern (INOUT) mit zusammengesetzten oder anwenderdefinierten Datentyp wird in der Spalte "Aktualwert" ein Zeiger auf den Datentyp angezeigt.
Anfangswert	Hierbei handelt es sich um den Anfangswert, den Sie für die Variable festgelegt haben, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Aktualwert	Offline: Hierbei handelt es sich um den Wert, den die Variable beim Öffnen des Datenbausteins bzw. nach Ihrer letzten gespeicherten Änderung hatte (auch wenn Sie den DB online geöffnet haben, wird diese Anzeige nicht aktualisiert!). Online: Der aktuelle Wert beim Öffnen des Datenbausteins wird angezeigt, jedoch nicht automatisch aktualisiert. Zum Aktualisieren der Anzeige drücken Sie die Taste F5. Dieses Feld können Sie editieren, wenn es nicht zu einem Durchgangsparameter (INOUT) mit einem zusammengesetzten oder anwenderdefinierten Datentyp gehört. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein.
Kommentar	Hierbei handelt es sich um den Kommentar, der in der zur Dokumentation der Variable vergeben wurde. Dieses Feld können Sie in der Datensicht nicht bearbeiten.

10.4 Eingeben und Speichern in Datenbausteinen

10.4.1 Eingeben der Datenstruktur von globalen Datenbausteinen

Haben Sie einen Datenbaustein geöffnet, der keinem UDT oder FB zugeordnet ist, so können Sie in der Deklarationssicht des Datenbausteins seine Struktur festlegen. Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

1. Öffnen Sie einen globalen Datenbaustein, d.h. einen Baustein, der keinem UDT oder FB zugeordnet ist.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird
3. Legen Sie die Struktur fest, indem Sie die angezeigte Tabelle anhand der nachstehenden Angaben ausfüllen.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangs-wert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Defaultwert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert für die Variable festlegen.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variable eingeben. Der Kommentar darf 80 Zeichen lang sein.

10.4.2 Eingeben / Anzeige der Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)

Eingabe

Wenn Sie einen Datenbaustein einem FB zuordnen (Instanz-DB), dann definiert die Variablendeklaration des FBs die Struktur des DBs. Änderungen sind nur im zugeordneten FB möglich:

1. Öffnen Sie den zugeordneten Funktionsbaustein.
2. Bearbeiten Sie die Variablendeklarationstabelle des Funktionsbausteins.
3. Legen Sie den Instanz-Datenbaustein erneut an!

Anzeige

Sie können sich in der Deklarationssicht des Instanz-DBs anzeigen lassen, wie die Variablen im FB deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluß.

Bei nicht globalen Datenbausteinen kann die Deklarationssicht nicht verändert werden.

Spalte	**Erläuterung**
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Deklaration	Diese Spalte zeigt an, wie die Variablen in der Variablendeklaration des FBs vereinbart worden sind: • Eingangsparameter (IN) • Ausgangsparameter (OUT) • Durchgangsparameter (IN_OUT) • statische Daten (STAT) Die deklarierten temporären Daten des FBs sind nicht im Instanz-Datenbaustein.
Name:	Anzeige des Namens, der durch die Variablendeklaration des FBs für die Variable festgelegt ist.
Typ	Anzeige des Datentyps, der durch die Variablendeklaration des FBs für die Variable festgelegt ist. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben. Werden innerhalb des FBs weitere Funktionsbausteine aufgerufen, für deren Aufruf statische Variablen vereinbart worden sind, dann kann hier auch ein FB oder Systemfunktionsbaustein (SFB) als Datentyp angegeben sein.
Anfangs-wert:	Hierbei handelt es sich um den Vorbelegungswert, den Sie in der Variablendeklaration des FB für die Variable festgelegt haben, wenn die Software nicht den Defaultwert des Datentyps übernehmen sollte. Der Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Kommentar	Hierbei handelt es sich um den Kommentar, der in der Variablendeklaration des FBs zur Dokumentation des Datenelements vergeben wurde. Dieses Feld können Sie nicht bearbeiten.

Hinweis

Bei Datenbausteinen, die einem FB zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

10.4.3 Eingeben der Struktur von anwenderdefinierten Datentypen (UDT)

1. Öffnen Sie den anwenderdefinierten Datentyp (UDT).
2. Lassen Sie sich die Deklarationssicht anzeigen, falls diese nicht bereits angezeigt wird.
3. Legen Sie die Struktur des anwenderdefinierten Datentyps fest, indem Sie die Reihenfolge der Variablen, ihren Datentyp und ggf. einen Anfangswert anhand der nachstehenden Angaben festlegen.
4. Sie beenden die Eingabe einer Variablen, indem Sie die Zeile mit der TAB-oder RETURN-Taste verlassen.

Spalte	Erläuterung
Adresse:	Anzeige der Adresse, die STEP 7 für die Variable automatisch zuordnet, wenn Sie die Eingabe einer Deklaration beenden.
Name	Geben Sie hier den Namen ein, den Sie jeder Variablen zuweisen müssen.
Typ	Geben Sie hier den Datentyp der Variablen ein (BOOL, INT, WORD, ARRAY, usw.). Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder wiederum anwenderdefinierte Datentypen haben.
Anfangs-wert	Geben Sie hier den Anfangswert ein, wenn die Software nicht den Default-Wert für den eingegebenen Datentyp übernehmen soll. Alle eingegebenen Werte müssen kompatibel zu den Datentypen sein. Der Anfangswert wird beim erstmaligen Speichern einer Instanz des UDTs (einer Variablen bzw. eines Datenbausteins) für die Variable als aktueller Wert übernommen, sofern Sie nicht explizit einen anderen aktuellen Wert vorgeben.
Kommentar	In diesem Feld können Sie einen Kommentar zur Dokumentation der Variablen eingeben. Der Kommentar darf 80 Zeichen lang sein.

10.4.4 Eingeben / Anzeige der Struktur von Datenbausteinen mit zugeordnetem UDT

Eingabe

Wenn Sie einen Datenbaustein einem UDT zuordnen, dann definiert die Datenstruktur des UDT die Struktur des DB. Änderungen sind nur im zugeordneten UDT möglich:

1. Öffnen Sie den anwenderdefinierten UDT.
2. Bearbeiten Sie die Struktur des anwenderdefinierten Datentyps.
3. Legen Sie den Datenbaustein neu an!

Anzeige

Sie können sich in der Deklarationssicht des DBs nur anzeigen lassen, wie die Variablen im UDT deklariert wurden.

1. Öffnen Sie den Datenbaustein.
2. Lassen Sie sich die Deklarationssicht des Datenbausteins anzeigen, falls diese nicht bereits angezeigt wird.
3. Erläuterungen zur angezeigten Tabelle finden Sie im Anschluß.

Die Deklarationssicht kann nicht verändert werden. Änderungen sind nur im zugeordneten UDT möglich.

Spalte	Erläuterung
Adresse	Anzeige der Adresse, die STEP 7 für die Variable automatisch vergibt.
Name	Anzeige des Namens, der durch den UDT für die Variable festgelegt ist.
Typ:	Anzeige der im UDT festgelegten Datentypen. Die Variablen können elementare Datentypen, zusammengesetzte Datentypen oder anwenderdefinierte Datentypen haben.
Anfangswert	Anzeige des Vorbelegungswerts, den Sie im UDT für die Variable festgelegt haben, wenn die Software nicht den Default-Wert des Datentyps übernehmen sollte. Dieser Anfangswert wird beim erstmaligen Speichern des Datenbausteins für die Variable als aktueller Wert übernommen, falls Sie nicht explizit einen aktuellen Wert der Variable festlegen.
Kommentar	Hierbei handelt es sich um den Kommentar, der durch den UDT zur Dokumentation des Datenelements festgelegt ist.

Hinweis

Bei Datenbausteinen, die einem UDT zugeordnet sind, können Sie nur die aktuellen Werte für die Variablen bearbeiten. Das Eingeben von aktuellen Werten für die Variablen erfolgt in der Datensicht von Datenbausteinen.

10.4.5 Ändern von Datenwerten in der Datensicht

Das Bearbeiten von aktuellen Werten ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Geben Sie die gewünschten aktuellen Werte für die Datenelemente in den Feldern der Spalte "Aktualwert" eingeben. Die Aktualwerte müssen kompatibel zum Datentyp der Datenelemente sein.

Fehlerhafte Eingaben bei der Bearbeitung werden sofort erkannt und rot dargestellt (z.B. wenn ein eingegebener aktueller Wert nicht kompatibel zum Datentyp ist). Diese Fehler müssen vor dem Speichern des Datenbausteins behoben sein.

Hinweis

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen gesichert.

10.4.6 Rücksetzen von Datenwerten auf die Anfangswerte

Das Rücksetzen ist nur in der Datensicht von Datenbausteinen möglich.

1. Schalten Sie ggf. mit dem Menübefehl **Ansicht > Datensicht** die Tabellendarstellung in die Datensicht um.
2. Wählen Sie hierfür den Menübefehl **Bearbeiten > Datenbaustein** initialisieren.

Alle Variablen werden wieder mit ihrem vorgesehenen Anfangswert versorgt, d.h. die aktuellen Werte aller Variablen werden durch den jeweiligen Anfangswert überschrieben.

Hinweis

Änderungen der Datenwerte werden erst beim Speichern von Datenbausteinen gesichert.

10.4.7 Speichern von Datenbausteinen

Um neu erstellte Datenbausteine bzw. Änderungen bei Datenwerten in Datenbausteinen in die PG-Datenhaltung zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Dazu werden die Daten auf die Festplatte des PGs geschrieben.

Speichern von Bausteinen auf die Festplatte des PGs:

1. Aktivieren Sie das Arbeitsfenster des zu speichernden Bausteins.
2. Wählen Sie:
   - den Menübefehl **Datei > Speichern**, wenn Sie den Baustein unter gleichem Namen abspeichern möchten.
   - den Menübefehl **Datei > Speichern** unter, wenn Sie den Baustein unter einem anderen S7-Anwenderprogramm bzw. unter anderem Namen abspeichern möchten. Geben Sie im folgenden Dialogfeld den neuen Pfad bzw. den neuen Baustein an. Sie dürfen als DB-Nummer nicht DB0 vergeben, da diese Nummer für das System reserviert ist.

In beiden Fällen wird der Baustein nur bei fehlerloser Syntax gespeichert. Syntaktische Fehler werden bei der Erstellung sofort erkannt und rot dargestellt. Diese Fehler müssen vor dem Speichern des Bausteins behoben sein.

Hinweis

- Das Speichern von Bausteinen oder Quellen unter anderen Projekten oder Bibliotheken können Sie auch im SIMATIC Manager durchführen (z.B. durch Drag&Drop).
- Das Speichern von Bausteinen oder kompletten Anwenderprogrammen auf eine Memory Card wird nur im SIMATIC Manager durchgeführt.
- Wenn beim Speichern oder Übersetzen großer Bausteine Probleme auftreten, sollten Sie das Projekt reorganisieren. Rufen Sie dazu den Menübefehl **Datei > Reorganisieren** im SIMATIC Manager auf. Versuchen Sie anschließend das Speichern bzw. Übersetzen erneut.

---

11 Erstellen von AWL-Quellen

11.1 Grundlagen zum Programmieren in AWL-Quellen

Sie haben die Möglichkeit, Ihr Programm bzw. Teile davon, als AWL-Quelle einzugeben und diese anschließend in Bausteine zu übersetzen. Die Quelle kann den Code für mehrere Bausteine enthalten, die anschließend in einem Lauf als Bausteine übersetzt werden.

Das Erstellen des Programms über eine Quelle hat folgende Vorteile:

- Sie können Ihre Quelle mit beliebigen ASCII-Editoren erstellen und bearbeiten, anschließend importieren und mit dieser Applikation in einzelne Bausteine übersetzen. Durch das Übersetzen werden die einzelnen Bausteine erzeugt und im S7-Anwenderprogramm abgelegt.
- Sie können mehrere Bausteine in einer Quelle programmieren.
- Sie können eine Quelle trotz Syntaxfehler abspeichern. Das ist bei der Erstellung von Codebausteinen mit inkrementeller Syntaxprüfung nicht möglich. Allerdings werden Sie erst beim Übersetzen der Quelle auf Syntaxfehler aufmerksam gemacht.

Die Quelle wird in der Syntax der Programmiersprache "Anweisungsliste (AWL)" erstellt. Die Strukturierung der Quelle in Bausteine, Variablendeklaration oder Netzwerke erfolgt über Schlüsselwörter.

Beim Erstellen von Bausteinen in AWL-Quellen sind zu beachten:

- Regeln zum Programmieren von AWL-Quellen
- Zulässige Syntax und Formate in AWL-Quellen
- Zulässige Bausteinstruktur von AWL-Quellen

11.2 Regeln zum Programmieren in AWL-Quellen

11.2.1 Regeln zur Eingabe von Anweisungen in AWL-Quellen

Eine AWL-Quelle besteht prinzipiell aus fortlaufendem Text. Damit sie sich in Bausteine übersetzen läßt, müssen bestimmte Strukturen und Syntaxvorschriften eingehalten werden.

Für die Erstellung von Anwenderprogrammen als AWL-Quelle gelten folgende allgemeine Regeln:

Thema	**Regel**
Syntax	Die Syntax der AWL-Anweisungen ist die gleiche wie im inkrementellen AWL-Editor. Eine Ausnahme besteht beim Aufrufbefehl CALL.
CALL	In einer Quelle geben Sie die Parameter in Klammern an. Die einzelnen Parameter werden dabei durch ein Komma voneinander getrennt. Beispiel Aufruf FC (einzeilig): CALL FC 10 (param1 :=E0.0,param2 :=E0.1); Beispiel Aufruf FB (einzeilig): CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1); Beispiel Aufruf FB (mehrzeilig): CALL FB10, DB100 (para1 :=E0.0, para2 :=E0.1); Hinweis: Übergeben Sie im ASCII-Editor bei einem Bausteinaufruf die Parameter in definierter Reihenfolge. Ansonsten stimmen u. U. die Kommentarzuordnung dieser Zeilen bei AWL- und Quell-Ansicht nicht überein.
Groß- oder Kleinschreibung	Der Editor dieser Applikation berücksichtigt allgemein keine Groß- oder Kleinschreibung. Ausgenommen davon sind Systemattribute und Sprungmarken. Bei Eingabe von Zeichenketten (Datentyp STRING) muß ebenfalls auf Groß- und Kleinschreibung geachtet werden.Schlüsselwörter werden in Großbuchstaben dargestellt. Beim Übersetzen wird Gross- und Kleinschreibung jedoch nicht berücksichtigt, so daß Sie Schlüsselwörter in Groß- oder Kleinbuchstaben oder gemischt angeben können.
Semikolon	Kennzeichnen Sie das Ende jeder AWL-Anweisung und jeder Variablendeklaration mit einem Semikolon. Sie können mehr als eine Anweisung pro Zeile eingeben.
zwei Schrägstriche (//)	Beginnen Sie jeden Kommentar mit zwei Schrägstrichen (//) und beenden Sie jede Kommentareingabe mit der RETURN-Taste.

11.2.2 Regeln für Variablendeklaration in AWL-Quellen

Zu jedem Baustein der Quelle müssen Sie die entsprechenden Variablen deklarieren.

Die Variablendeklaration steht vor dem Anweisungsteil des Bausteins.

Die Variablen müssen - soweit vorhanden - in der Reihenfolge der Deklarationstypen deklariert werden. Damit stehen alle Variablen eines Deklarationstyps zusammen.

Während Sie in KOP, FUP oder AWL eine entsprechende Variablendeklarationstabelle ausfüllen, müssen Sie hier mit entsprechenden Schlüsselwörtern arbeiten.

Schlüsselwörter für die Variablendeklaration

Deklarationstyp	Schlüsselwörter	Gültig für ...
Eingangsparameter	"VAR_INPUT" Deklarationsliste "END_VAR"	FBs, FCs
Ausgangsparameter	"VAR_OUTPUT" Deklarationsliste "END_VAR"	FBs, FCs
Durchgangsparameter	"VAR_IN_OUT" Deklarationsliste "END_VAR"	FBs, FCs
Statische Variablen	"VAR" Deklarationsliste "END_VAR"	FBs
Temporäre Variablen	"VAR_TEMP" Deklarationsliste" END_VAR"	OBs, FBs, FCs

Das Schlüsselwort END_VAR kennzeichnet das Ende einer Deklarationsliste.

Die Deklarationsliste ist Liste der Variablen eines Deklarationstyps. Darin können den Variablen Vorbelegungswerte zugeordnet werden (Ausnahme: VAR_TEMP). Das folgende Beispiel zeigt den Aufbau eines Eintrags in der Deklarationsliste:

Dauer_Motor1 : S5TIME := S5T#1H_30M ;

Variable Datentyp Vorbelegungswert

Hinweis

- Der Variablenname muß mit einem Buchstaben oder einem Unterstrich beginnen. Sie dürfen keinen Variablennamen vergeben, der einem der reservierten Schlüsselwörter entspricht.
- Sind Variablennamen in den lokalen Deklarationen und in der Symboltabelle identisch, dann kennzeichnen Sie lokale Variablen mit einem # vor dem Namen und setzen Sie Variablen der Symboltabelle in Anführungszeichen. Andernfalls interpretiert der Baustein die Variable als lokale Variable.

11.2.3 Regeln zur Reihenfolge der Bausteine in AWL-Quellen

Aufgerufene Bausteine stehen vor den aufrufenden Bausteinen, d.h.:

- Der in den häufigsten Fällen verwendete OB1, der andere Bausteine aufruft, steht zuletzt. Bausteine, die wiederum von den aus OB1 aufgerufenen Bausteinen aufgerufen werden, müssen vor diesen stehen.
- Anwenderdefinierte Datentypen (UDTs) stehen vor den Bausteinen, in denen Sie verwendet werden.
- Datenbausteine mit zugeordnetem anwenderdefinierten Datentyp (UDT) stehen hinter dem UDT.
- Globale Datenbausteine stehen vor allen Bausteinen, aus denen Sie aufgerufen werden.
- Ein Instanz-Datenbausteine steht hinter dem zugeordneten Funktionsbaustein.
- Der DB 0 ist vorbelegt. Sie können also keinen DB dieses Namens erzeugen.

11.2.4 Regeln zum Festlegen von Systemattributen in AWL-Quellen

Systemattribute können Bausteinen und Parametern zugewiesen werden. Sie steuern die Meldungs- und Verbindungsprojektierung, Bedien- und Beobachtungsfunktionen und die Leittechnikprojektierung.

Für die Eingabe in der Quelle gilt folgendes:

- Die Schlüsselwörter für Systemattribute beginnen immer mit S7_.
- Die Systemattribute stehen in geschweiften Klammern.
- Syntax: {S7_idenifier := 'string'}
  Mehrere Identifier sind durch ";" getrennt.
- Systemattribute für Bausteine stehen vor den Bausteineigenschaften und nach dem Schlüsselworten ORGANIZATION_ bzw. TITLE.
- Systemattribute für Parameter stehen in der Parameterdeklaration, d.h. vor dem Doppelpunkt der Datendeklaration.
- Es wird zwischen Groß- und Kleinbuchstaben unterschieden, d.h. die Groß-und Kleinschreibung ist bei der Eingabe von Systemattributen relevant!

Die Systemattribute für Bausteine können bei der inkrementellen Eingabe mit dem Menübefehl **Datei > Eigenschaften** im Register "Attribute" kontrolliert bzw. verändert werden.

Die Systemattribute für Parameter können bei der inkrementellen Eingabe mit dem Menübefehl **Bearbeiten > Objekteigenschaften** kontrolliert bzw. verändert werden. Dabei muß der Cursor im Namensfeld der Parameterdeklaration stehen.

11.2.5 Regeln zum Festlegen von Bausteineigenschaften in AWL-Quellen

Mit Hilfe von Bausteineigenschaften können Sie die erstellten Bausteine besser identifizieren (z.B. bei der Versionspflege) oder Sie schützen Bausteine vor unberechtigten Veränderungen.

Die Bausteineigenschaften können bei der inkrementellen Eingabe mit dem Menübefehl **Datei > Eigenschaften** in den Registern "Allgemein Teil 1" und "Allgemein Teil 2" kontrolliert bzw. verändert werden.

Alle anderen Bausteineigenschaften können nur in der Quelle eingegeben werden.

Dabei gilt in der Quelle folgendes:

- Bausteineigenschaften werden vor dem Variablendeklarationsteil angegeben.
- Jede Bausteineigenschaft steht in einer eigenen Zeile.
- Am Ende der Zeile steht kein Semikolon.
- Die Bausteineigenschaften werden durch die Verwendung von Schlüsselwörtern angegeben.
- Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge dieser nach Tabelle der Bausteineigenschaften einzuhalten.
- Die jeweils möglichen Bausteineigenschaften einer Bausteinart sind der Zuordnung Bausteineigenschaften zu Bausteinarten zu entnehmen

---

Hinweis

Die Bausteineigenschaften werden auch im SIMATIC Manager in den Objekteigenschaften für einen Baustein angezeigt. Dort können auch die Eigenschaften AUTHOR, FAMILY, NAME und VERSION editiert werden.

Bausteineigenschaften und Reihenfolge

Wenn Sie Bausteineigenschaften eingeben möchten, dann ist die Reihenfolge der Eingabe nach folgender Tabelle zu beachten.

Reihen-folge	Schlüsselwort / Eigenschaft	Bedeutung	Beispiel
1.	[KNOW_HOW_PROTECT]	Bausteinschutz; ein Baustein, der mit dieser Option übersetzt wurde, läßt keinen Einblick in den Anweisungsteil zu.	KNOW_HOW_PROTECT
2.	[AUTHOR:]	Name des Autors, Firmenname, Abteilungsname od. andere Namen (max. 8 Zeichen, ohne Leerzeichen)	AUTHOR : Siemens, aber kein Schlüsselwort
3.	[FAMILY:]	Name der Bausteinfamilie: z.B. Regler. (max. 8 Zeichen, ohne Leerzeichen).	FAMILY : Regler, aber kein Schlüsselwort
4.	[NAME:]	Bausteinname (max. 8 Zeichen)	NAME : PID, aber kein Schlüsselwort
5.	[VERSION: int1 . int2]	Versionsnummer des Bausteins (beide Zahlen zwischen 0..15, d.h. 0.0 - 15.15))	VERSION : 3.10
6.	[CODE_VERSION1]	Kennung, ob ein FB multiinstanzfähig ist oder nicht. Wenn Sie Multiinstanzen deklarieren möchten, darf der FB dieses Eigenschaft nicht haben.	CODE_VERSION1
7.	[UNLINKED] nur für DBs!	Ein Datenbaustein mit der Eigenschaft UNLINKED wird nicht in das Programm eingebunden.	
8.	[READ_ONLY] nur für DBs!	Schreibschutz für Datenbausteine; seine Daten können nur gelesen u. nicht verändert werden.	FAMILY= Beispiele VERSION= 3.10 READ_ONLY

11.2.6 Zulässige Bausteineigenschaften je Bausteinart

Die nachfolgenden Tabelle zeigt, welche Bausteineigenschaften bei welchen Bausteinarten vereinbart werden können.

Eigenschaft	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	●	●	●	●	–
AUTHOR	●	●	●	●	–
FAMILY	●	●	●	●	–
NAME	●	●	●	●	–
VERSION	●	●	●	●	–
UNLINKED	–	–	–	●	–
READ_ONLY	–	–	–	●	–

Bausteinschutz einrichten mit KNOW_HOW_PROTECT

Sie können für Bausteine einen Bausteinschutz einrichten, indem Sie das Schlüsselwort KNOW_HOW_PROTECT bei der Programmierung des Bausteins in der AWL-Quelle angeben.

Der Bausteinschutz hat folgende Konsequenzen:

- Wenn Sie später einen übersetzten Baustein im inkrementellen AWL-, FUP- oder KOP-Editor ansehen wollen, kann der Anweisungsteil des Bausteins nicht eingesehen werden.
- In der Variablendeklarationsliste des Bausteins werden nur die Variablen der Deklarationstypen var_in, var_out und var_in_out angezeigt. Die Variablen der Deklarationstypen var_stat und var_temp bleiben verborgen.
- Das Schlüsselwort KNOW_HOW_PROTECT wird vor allen anderen Bausteineigenschaften eingegeben.

Schreibschutz auf Datenbausteine einrichten mit READ_ONLY

Für Datenbausteine können Sie einen Schreibschutz einrichten, so daß sie beim Programmablauf nicht überschrieben werden können. Der Datenbaustein muß dazu als AWL-Quelle vorliegen.

Geben Sie in der Quelle das Schlüsselwort READ_ONLY ein. Dieses muß direkt vor den Variablendeklarationen in einer eigenen Zeile stehen.

11.3 Struktur von Bausteinen in AWL-Quellen

11.3.1 Struktur von Bausteinen in AWL-Quellen

Die Strukturierung von Bausteinen in AWL-Quelle erfolgt über Schlüsselwörter. Abhängig von der Art des Baustein unterscheidet man die Struktur von

- Codebausteinen
- Datenbausteinen
- anwenderdefinierten Datentypen

11.3.2 Struktur von Codebausteinen in AWL-Quellen

Ein Codebaustein besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter gekennzeichnet werden:

- Bausteinanfang,
- gekennzeichnet durch Schlüsselwort und Bausteinnummer oder Bausteinnamen, also z.B.:
  - "ORGANIZATION_BLOCK OB 1" für einen Organisationsbaustein.
  - "FUNCTION_BLOCK FB6" für einen Funktionsbaustein.
  - "FUNCTION FC 1 : INT" für eine Funktion.Bei Funktionen wird zusätzlich der Funktionstyp angegeben. Dieser kann von elementarem oder zusammengesetztem Datentyp (außer ARRAY und STRUCT) sein und bestimmt den Datentyp des Rückgabewertes (RET_VAL). Soll kein Wert zurückgegeben werden, ist das Schlüsselwort VOID anzugeben.
- Optionaler Bausteintitel, eingeleitet mit dem Schlüsselwort "TITLE (max. Eingabe des Titels: 64 Zeichen).
- Weiterer Kommentar, beginnend mit zwei Schrägstrichen // am Zeilenanfang.
- Eintragung der Bausteineigenschaften (optional).
- Variablendeklarationsteil.
- Anweisungsteil, beginnend mit BEGIN. Der Anweisungsteil besteht wiederum aus ein oder mehreren Netzwerken, die mit NETWORK gekennzeichnet werden. Die Angabe einer Netzwerknummer ist nicht erlaubt.
- Optionaler Netzwerktitel für jedes implementierte Netzwerk, eingeleitet mit dem Schlüsselwort "TITLE =" (max. Eingabe des Titels: 64 Zeichen).
- Weitere Kommentare je Netzwerk, beginnend mit zwei Schrägstrichen "//" am Zeilenanfang.
- Bausteinende, gekennzeichnet durch END_ORGANIZATION_BLOCK, END_FUNCTION_BLOCK oder END_FUNCTION.
- Zwischen Bausteinart und Bausteinnummer steht ein Leerzeichen. Der symbolische Bausteinname kann durch Anführungszeichen gekennzeichnet werden, um die Eindeutigkeit zwischen Namen lokaler Variablen und Namen der Symboltabelle sicherzustellen.

11.3.3 Struktur von Datenbausteinen in AWL-Quellen

Ein Datenbaustein besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter eingeleitet werden:

- Bausteinanfang, gekennzeichnet durch Schlüsselwort und Bausteinnummer oder Bausteinnamen, also z.B. DATA_BLOCK DB 26
- Angabe des zugeordneten UDTs oder des FBs, dem der DB zugeordnet ist (optional).
- Optionaler Bausteintitel, eingeleitet mit dem Schlüsselwort "TITLE =" (Eingaben > 64 Zeichen werden abgeschnitten).
- Optionaler Bausteinkommentar, beginnend mit "//".
- Eintragung der Bausteineigenschaften (optional)
- Variablendeklarationsteil (optional)
- Zuweisungsteil mit Vorbelegungswerten, beginnend mit BEGIN (optional)
- Bausteinende, gekennzeichnet durch END_DATA_BLOCK.

Es gibt drei Typen von Datenbausteinen:

- Datenbausteine, benutzerdefiniert
- Datenbausteine, mit zugeordnetem anwenderdefinierten Datentyp (UDT)
- Datenbausteine, die einem FB zugeordnet sind (Instanz-DB)

11.3.4 Struktur von anwenderdefinierten Datentypen in AWL-Quellen

Ein anwenderdefinierter Datentyp besteht aus folgenden Bereichen, die jeweils durch entsprechende Schlüsselwörter eingeleitet werden:

- Bausteinanfang, gekennzeichnet durch Schlüsselwort TYPE und Nummer oder Namen, also z.B. TYPE UDT 20
- Angabe eines strukturierten Datentyps
- Bausteinende, gekennzeichnet durch END_TYPE.

Bei der Eingabe von anwenderdefinierten Datentypen müssen Sie beachten, daß sie vor den Bausteinen stehen, in denen Sie verwendet werden.

11.4 Syntax und Formate für Bausteine in AWL-Quellen

11.4.1 Syntax und Formate für Bausteine in AWL-Quellen

Die Formattabellen stellen die Syntax und die Formate dar, die Sie bei der Programmierung von AWL-Quellen beachten müssen. Die Syntax ist folgendermaßen dargestellt:

- Jedes Element ist in der rechten Spalte beschrieben.
- Unbedingt erforderliche Elemente werden in Anführungszeichen dargestellt.
- Die eckigen Klammern [...] bedeuten, daß der Inhalt dieser Klammern nicht unbedingt angegeben werden muß.
- Schlüsselwörter werden in Großbuchstaben angegeben.

11.4.2 Formattabelle von OBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Organisationsbausteine in einer AWL-Quelle:

Aufbau	**Beschreibung**
"ORGANIZATION_BLOCK" ob_nr oder ob_name	ob_nr ist die Bausteinnummer, z.B. OB 1; ob_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten.
[Bausteinkommentar]	Kommentare können hinter "//" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von temporären Variablen
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerkes
[TITLE=]	Netzwerktitel (max. 64 Zeichen).
[Netzwerkkommentar]	Kommentare können hinter "//" eingegeben werden.
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_ORGANIZATION_BLOCK"	Schlüsselwort zum Beenden des Organisationsbausteins

11.4.3 Formattabelle von FBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Funktionsbausteine in einer AWL-Quelle:

Aufbau	Beschreibung
"FUNCTION_BLOCK" fb_nr oder fb_name	fb_nr ist die Bausteinnummer, z.B. FB 6; fb_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten.
[Bausteinkommentar]	Kommentare können hinter "//" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von Eingangs-, Ausgangs- und Durchgangsparametern sowie von temporären od. statischen Variablen. Die Deklaration der Parameter enthält ggf. die Deklarationen der Systemattribute für Parameter
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerkes
[TITLE=]	Netzwerktitel (max. 64 Zeichen).
[Netzwerkkommentar]	Kommentare können hinter "//" eingegeben werden.
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_FUNCTION_BLOCK	Schlüsselwort zum Beenden des Funktionsbausteins

11.4.4 Formattabelle von FCs

Nachfolgende Tabelle zeigt in Kurzform das Format für Funktionen in einer AWL-Quelle:

Aufbau	Beschreibung
"FUNCTION" fc_nr : fc_type oder fc_name : fc_type	fc_nr ist die Bausteinnummer, z.B. FC 5; fc_name ist der symbolische Name des Bausteins laut Symboltabelle fc_type gibt den Datentyp des Rücksprungwerts (RET_VAL) der Funktion an. Es kann sich um einen elementaren oder strukturierten Datentyp (außer ARRAY und STRUCT) oder um VOID handeln. Wollen Sie Systemattribute für den Rücksprungwert (RET_VAL) verwenden, dann müssen Sie die Systemattribute für Parameter vor den Dopppelpunkt der Datendeklaration eintragen.
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten.

Aufbau	Beschreibung
[Bausteinkommentar]	Kommentare können hinter "//" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Variablendeklarationsteil	Deklaration von Eingangs-, Ausgangs- und Durchgangsparametern sowie von temporären Variablen
"BEGIN"	Schlüsselwort zur Trennung der AWL-Anweisungen vom Variablendeklarationsteil
NETWORK	Beginn eines Netzwerkes
[TITLE=]	Netzwerktitel (max. 64 Zeichen).
[Netzwerkkommentar]	Kommentare können hinter "//" eingegeben werden.
Liste der AWL-Anweisungen	Bausteinanweisungen
"END_FUNCTION"	Schlüsselwort zum Beenden der Funktion

11.4.5 Formattabelle von DBs

Nachfolgende Tabelle zeigt in Kurzform das Format für Datenbausteine in einer AWL-Quelle:

Aufbau	Beschreibung
"DATA_BLOCK" db_nr oder db_name	db_nr ist die Bausteinnummer, z.B. DB 5; db_name ist der symbolische Name des Bausteins laut Symboltabelle
[TITLE=]	Kommentar bis <RETURN>; Eingaben > 64 Zeichen werden abgeschnitten.
[Bausteinkommentar]	Kommentare können hinter "//" eingegeben werden
[Systemattribute für Bausteine]	Systemattribute für Bausteine
[Bausteineigenschaften]	Bausteineigenschaften
Deklarationsteil	Angabe von UDT oder FB, auf den sich der Baustein bezieht, als Bausteinnummer oder Name laut Symboltabelle oder Angabe des zusammengesetzten Datentyps.
"BEGIN"	Schlüsselwort zum Trennen des Deklarationsteils von der Liste der Wertzuweisungen.
[Zuweisung von Anfangswerten]	Variablen können bestimmte Anfangswerte zugewiesen werden. Dabei werden einzelnen Variablen Konstanten zugeordnet oder es wird auf andere Bausteine verwiesen.
"END_DATA_BLOCK"	Schlüsselwort zum Beenden des Datenbausteins

11.5 Erstellen von AWL-Quellen

11.5.1 Anlegen von AWL-Quellen

Die Quelle muß unter dem S7-Programm in einem Quell-Ordner angelegt werden. Sie können Quellen im SIMATIC Manager oder im Editor-Fenster anlegen.

Anlegen von Quellen im SIMATIC Manager

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Wählen Sie zum Einfügen einer AWL-Quelle den Menübefehl **Einfügen > S7-Software > AWL-Quelle**.

Anlegen von Quellen im Editor-Fenster

1. Wählen Sie den Menübefehl **Datei > Neu**.
2. Wählen Sie im Dialogfeld den Quell-Ordner desselben S7-Programms, in dem sich auch das Anwenderprogramm mit den Bausteinen befindet.
3. Weisen Sie der anzulegenden Quelle einen Namen zu.
4. Bestätigen Sie mit "OK".

Die Quelle wird unter dem von Ihnen vergebenen Namen angelegt und in einem Arbeitsfenster zur weiteren Bearbeitung angezeigt.

11.5.2 Bearbeiten von S7-Quellen

In welcher Programmiersprache und mit welchem Editor die jeweilige Quelle bearbeitet wird, ist in den Objekteigenschaften der Quelle festgelegt. Damit wird für die Bearbeitung immer der richtige Editor mit der entsprechenden Programmiersprache gestartet. Das STEP 7-Basispaket unterstützt die Programmierung in AWL-Quellen.

Andere Programmiersprachen stehen Ihnen jeweils als Optionspaket zur Verfügung. Nur wenn die jeweilige Optionssoftware auf Ihrem Rechner geladen ist, können Sie durch Doppelklick auf die Quelle den entsprechenden Editor starten.

Gehen Sie folgendermaßen vor:

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Starten Sie den für die Bearbeitung notwendigen Editor wie folgt:
   - Führen Sie einen Doppelklick auf die jeweilige Quelle in der rechten Fensterhälfte aus.
   - Markieren Sie die entsprechende Quelle in der rechten Fensterhälfte und rufen Sie den Menübefehl **Bearbeiten > Objekt öffnen** auf.

11.5.3 Einfügen von Bausteinvorlagen in AWL-Quellen

Für das Programmieren in AWL-Quellen werden Ihnen Bausteinvorlagen für OBs, FBs, FCs, DBs, Instanz-DBs, DBs aus UDTs und UDTs angeboten. Die Bausteinvorlagen erleichtern Ihnen die Eingabe und Einhaltung von Syntax und Struktur.

Gehen Sie folgendermaßen vor:

1. Aktivieren Sie das Arbeitsfenster der Quelle, in die Sie eine Bausteinvorlage einfügen möchten.
2. Positionieren Sie den Cursor an die Stelle, hinter der die Bausteinvorlage eingefügt werden soll.
3. Wählen Sie den entsprechenden Menübefehl **Einfügen > Bausteinvorlage > OB/FB/FC/DB/IDB/DB** aus **UDT/UDT**.

Die Bausteinvorlage wird hinter der Cursorposition eingefügt.

11.5.4 Einfügen von externen Quellen

Sie können Ihre Quellen mit beliebigen ASCII-Editoren erstellen und bearbeiten, anschließend in ein Projekt importieren und in einzelne Bausteine übersetzen. Dazu müssen die Quellen in den Ordner "Quellen" des S7-Programms importiert werden, in dessen S7-Anwenderprogramm die aus der Übersetzung entstehenden Bausteine abgelegt werden sollen.

Gehen Sie folgendermaßen vor:

1. Markieren Sie den Ordner "Quellen" des S7-Programms, in den die externen Quellen importiert werden sollen.
2. Wählen Sie den Menübefehl **Einfügen > Externe Quelle**.
3. Geben Sie im anschließenden Dialogfeld die zu importierende Quelle an.

Der Dateiname der zu importierenden Quelle muß eine gültige Dateierweiterung besitzen. Aus der Dateierweiterung wird der Typ der Quelle in STEP 7 ermittelt. So entsteht z.B. aus einer Datei mit der Erweiterung **.awl** beim Importieren eine AWL-Quelle. Gültige Dateierweiterungen sind im Dialogfeld unter "Dateityp" ersichtlich.

Hinweis

Mit dem Menübefehl **Einfügen > Externe Quelle** können Sie auch Quellen einfügen, die Sie mit STEP 7 Version 1 erzeugt haben.

11.5.5 Generieren von AWL-Quellen aus Bausteinen

Sie haben die Möglichkeit, von bestehenden Bausteinen eine AWL-Quelle zu erstellen, die Sie mit jedem beliebigen Texteditor bearbeiten können. Die generierte Quelle befindet sich im Quell-Ordner desselben S7-Programms, aus dessen Anwenderprogramm die Bausteine ausgewählt werden.

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei > Quelle generieren**.
2. Wählen Sie im Dialogfeld den Quell-Ordner, in dem Sie die neue Quelle anlegen möchten.
3. Weisen Sie der Quelle in dem Textfeld einen Namen zu.
4. Wählen Sie im Dialogfeld "Auswahl STEP7-Baustein" den oder die Bausteine aus, die Sie in die zuvor angegebene Quelle generieren möchten. Die ausgewählten Bausteine werden im rechten Listenfeld angezeigt.
5. Bestätigen Sie mit "OK".

Aus den ausgewählten Bausteinen wird eine zusammenhängende AWL-Quelle generiert und Ihnen zur weiteren Bearbeitung in einem Arbeitsfenster angezeigt.

11.6 Speichern, Übersetzen von AWL-Quellen und Konsistenzprüfung

11.6.1 Speichern von AWL-Quellen

Sie können eine AWL-Quelle jederzeit im aktuellen Zustand abspeichern. Das Programm wird dabei nicht übersetzt, es findet keine Syntaxprüfung statt, d.h. Fehler werden mit abgespeichert.

Syntaxfehler werden erst beim Übersetzen der Quelle, bzw. bei einer Konsistenzprüfung festgestellt und gemeldet.

Quelle unter gleichem Namen speichern:

1. Aktivieren Sie das Arbeitsfenster der zu speichernden Quelle.
2. Wählen Sie den Menübefehl **Datei > Speichern**.

Quelle unter anderem Namen/in anderem Projekt speichern:

1. Aktivieren Sie das Arbeitsfenster der zu speichernden Quelle.
2. Wählen Sie den Menübefehl **Datei > Speichern unter**.
3. Wählen Sie im nachfolgenden Dialogfeld den Quell-Ordner an, dem die Quelle zugeordnet sein soll und geben Sie deren neuen Namen an.

11.6.2 Prüfen der Konsistenz in AWL-Quellen

Mit dem Menübefehl **Datei > Konsistenz prüfen** können Sie sich eventuelle Syntaxfehler in der AWL-Quelle anzeigen lassen. Dabei werden im Gegensatz zur Übersetzung keine Bausteine generiert.

Nach Abschluß der Konsistenzprüfung wird ein Dialogfeld aufgeblendet, das Ihnen die Gesamtzahl der gefundenen Fehler mitteilt.

Sind Fehler vorhanden, werden diese im unteren Teil des Arbeitsfensters einzeln mit Zeilenangabe aufgelistet. Diese müssen vor dem Übersetzen der Quelle behoben sein, damit alle Bausteine erzeugt werden.

11.6.3 Fehlersuche in AWL-Quellen

Das Arbeitsfenster für Quellen ist zweigeteilt. In der unteren Hälfte werden folgende Fehler aufgelistet:

- Fehler, die nach Anstoß eines Übersetzungsvorgangs mit dem **Menübefehl Datei > Übersetzen** festgestellt wurden.
- Fehler, die nach Anstoß der Konsistenzprüfung mit dem Menübefehl **Datei > Konsistenz prüfen** festgestellt wurden.

Um die Stelle eines Fehlers in der Quelle zu finden, positionieren Sie den Cursor auf die entsprechende Fehlermeldung in der unteren Fensterhälfte. Automatisch wird die entsprechende Textzeile in der oberen Hälfte markiert. Zusätzlich befindet sich die Fehlermeldung in der Statuszeile.

11.6.4 Übersetzen von AWL-Quellen

Voraussetzungen

Um das in einer Quelle erstellte Programm in Bausteine übersetzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Es können nur Quellen übersetzt werden, die im Ordner "Quellen" unter einem S7-Programm gespeichert sind.
- Parallel zum Ordner "Quellen" muß unter dem S7-Programm ein Ordner "Bausteine" liegen, in den die bei der Übersetzung entstehenden Bausteine gespeichert werden. Die in der Quelle programmierten Bausteine werden nur dann erzeugt, wenn die Quelle fehlerfrei übersetzt worden ist. Sind mehrere Bausteine in einer Quelle, werden nur die fehlerfreien erzeugt. Anschließend können Sie diese Bausteine einzeln öffnen, bearbeiten, in die CPU laden und testen.

Vorgehensweise im Editor

1. Öffnen Sie die Quelle, die Sie übersetzen wollen. Die Quelle muß im Quell-Ordner des S7-Programms liegen, in dessen S7-Anwenderprogramm die übersetzten Bausteine abgelegt werden sollen.
2. Wählen Sie den Menübefehl **Ansicht > Anzeigen mit > Symbolischer Darstellung**, damit in den übersetzten Bausteinen später die Symbolik angezeigt werden kann.
3. Wählen Sie den Menübefehl **Datei > Übersetzen**.
4. Sie erhalten das Dialogfeld "Übersetzungsprotokoll", das die Anzahl der übersetzten Zeilen und der gefundenen Syntaxfehler anzeigt.

Die in der Datei angegebenen Bausteine werden nur dann erzeugt, wenn die Quelle fehlerfrei übersetzt worden ist. Sind mehrere Bausteine in einer Quelle, werden nur die fehlerfreien erzeugt. Warnungen verhindern nicht das Erzeugen von Bausteinen.

Bei der Übersetzung erkannte Syntaxfehler werden in der unteren Hälfte des Arbeitsfensters dargestellt und müssen behoben werden, damit auch diese Bausteine erzeugt werden können.

Vorgehensweise im SIMATIC Manager

1. Öffnen Sie den entsprechenden Ordner "Quellen", in dem Sie darauf doppelklicken.
2. Wählen Sie eine oder mehrere Quelle(n) aus, die Sie übersetzen wollen. Sie können den Übersetzungslauf nicht für einen geschlossenen Ordner "Quellen" aufrufen, um damit alle enthaltene Quellen zu übersetzen.
3. Wählen Sie den Menübefehl **Datei > Übersetzen**, um die Übersetzung zu starten. Dabei wird je nach angewählter Quelle der richtige Übersetzer aufgerufen. Die erfolgreich übersetzten Bausteine werden danach im Ordner "Bausteine" unter dem S7-Programm abgelegt.
   Bei der Übersetzung erkannte Syntaxfehler werden in einem Dialogfeld dargestellt und müssen behoben werden, damit auch diese Bausteine erzeugt werden können.

11.7 Beispiele zu AWL-Quellen

11.7.1 Beispiele für Variablendeklarationen in AWL-Quellen

Variablen elementaren Datentyps

```
                // Kommentare werden durch zwei Schrägstriche vom Deklarationsteil getrennt.
VAR_INPUT       // Schlüsselwort Eingangsvariable
                in1 : INT;// Variablenname und Typ werden durch ":" getrennt
                in3 : DWORD;      // Jede Variablendeklaration wird durch ein Semikolon beendet.
                in2 : INT := 10;      // Optionale Festlegung des Anfangswertes in der Deklaration
END_VAR         // Abschluß der Deklaration von Variablen des gleichen Deklarationstyps
VAR_OUTPUT      // Schlüsselwort Ausgangsvariable
                out1 : WORD;
END_VAR         // Schlüsselwort Temporäre Variable
VAR_TEMP
                temp1 : INT;
END_VAR
```

Variablen vom Datentyp Feld

```
VAR_INPUT       // Eingangsvariable
                feld1 : ARRAY [1..20] of INT;           // feld1 ist ein eindimensionales Feld
                feld2 : ARRAY [1..20, 1..40] of DWORD;       // feld2 ist ein zweidimensionales Feld
END_VAR
```

Variablen vom Datentyp Struktur

```
VAR_OUT         // Ausgangsvariable
AUSGANG1:       STRUCT// AUSGANG1 ist vom Datentyp STRUCT
                 var1 : BOOL;       // Element 1 der Struktur
                 var2 : DWORD;    // Element 2 der Struktur
                 END_STRUCT;    // Ende der Struktur
END_VAR
```

11.7.2 Beispiel für OBs in AWL-Quellen

```
ORGANIZATION_BLOCK OB 1
TITLE = Beispiel fuer OB 1 mit unterschiedlichen Bausteinaufrufen
//Die 3 dargestellten Netzwerke zeigen Bausteinaufrufe
//mit und ohne Parameter

{S7_pdiag := 'true'}                                    //Systemattribut fuer Bausteine
AUTHOR          :       Siemens
FAMILY          :       Beispiel
NAME            :       Test_OB
VERSION         :       1.1
VAR_TEMP
Zwischenwert : INT;                     // Zwischenspeicher
END_VAR

BEGIN

NETWORK
TITLE = Aufruf einer Funktion mit Uebergabe von Parametern
// Einzeiliger Parameteruebergabe
CALL FC1 (param1 :=E0.0,param2 :=E0.1);

NETWORK
TITLE = Aufruf eines Funktionsbausteins mit
//Uebergabe von Parametern
// Mehrzeilige Parameterangabe
CALL  Ampelsteuerung, DB 6 (            // Name des FBs, Instanz-Datenbaustein
dur_g_p         := S5T#10S,             // Zuweisung aktueller Werte an Parameter

del_r_p         := S5T#30S,
starter         := TRUE,
t_dur_y_car     := T 2,
t_dur_g_ped     := T 3,
t_delay_y_car   := T 4,
t_dur_r_car     := T 5,
t_next_red_car  := T 6,
r_car           := "re_main",           // Anfuehrungszeichen kennzeichnen
y_car           := "ye_main",           // Namen der Symboltabelle
g_car           := "gr_main",
r_ped           := "re_int",
g_ped           := "gr_int");

NETWORK
TITLE = Aufruf eines Funktionsbausteins mit
//Uebergabe von Parametern
// Einzeilige Parameterangabe
CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);

END_ORGANIZATION_BLOCK
```

11.7.3 Beispiel für FCs in AWL-Quellen

```
FUNCTION FC 1: VOID
// Nur wegen Call!!
VAR_INPUT
  param1 : bool;
  param2 : bool;
END_VAR
begin
end_function


FUNCTION FC2 : INT
TITLE = Stueckzahlerhoehung
// Solange der uebergebene Wert < 1000 ist, erhoeht diese
//Funktion den uebergebenen Wert. Liegt die Stueckzahl
//ueber 1000, so wird ueber den Rueckgabewert der
//Funktion (RET_VAL) "-1" zurueckgeliefert.

AUTHOR      :     Siemens
FAMILY      :     Durchs
NAME        :     STUECK
VERSION     :     1.0

VAR_IN_OUT
STUECKZAHL : INT;                          // aktuell gefertigte Stueckzahl
END_VAR

BEGIN

NETWORK
TITLE = Erhoehung Stueckzahl um 1
// Solange die aktuelle Stueckzahl unter 1000 liegt,
// kann sie um 1 erhoeht werden
L STUECKZAHL; L 1000;                  // Beispiel fuer mehrere
>I; SPB ERR;                           // Anweisungen in einer Zeile.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;
END_FUNCTION
```

```
FUNCTION FC3 {S7_pdiag := 'true'} : INT
TITLE = Stueckzahlerhoehung
// Solange der uebergebene Wert < 1000 ist, erhoeht diese
//Funktion den uebergebenen Wert. Liegt die Stueckzahl
//ueber 1000, so wird ueber den Rueckgabewert der
//Funktion (RET_VAL) "-1" zurueckgeliefert.
//
//RET_VAL besitzt hier ein Systemattribut fuer Parameter
AUTHOR          :       SIEMENS
FAMILY          :       DURCH
NAME            :       STUECK
VERSION         :       1.0

VAR_IN_OUT
STUECKZAHL {S7_visible := 'true'}: INT;      // aktuell gefertigte Stueckzahl
//Systemattribute fuer Parameter
END_VAR

BEGIN

NETWORK
TITLE = Erhoehung Stueckzahl um 1
// Solange die aktuelle Stueckzahl unter 1000 liegt,
// kann sie um 1 erhoeht werden
L STUECKZAHL; L 1000;                       // Beispiel fuer mehrere
>I; SPB ERR;                                // Anweisungen in einer Zeile.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;

END_FUNCTION
```

11.7.4 Beispiel für FBs in AWL-Quellen

```
FUNCTION_BLOCK FB 6
TITLE = Ampelschaltung einfach
// Steuerung einer Ampelanlage fuer einen Fussgaengeruebergang
//an der Hauptstrasse

{S7_m_c := 'true'}          //Systemattribut fuer Bausteine
AUTHOR          :           SIEMENS
FAMILY          :           Ampel
NAME            :           Ampel01
VERSION         :           1.3

VAR_INPUT

starter         :           BOOL    :=      FALSE;  // Ueberquerungswunsch Fußgaenger
t_dur_y_car     :           TIMER;          // Zeitdauer Gruen Fussgaenger
t_next_r_car    :           TIMER;          // Zeitdauer zwischen Rot fuer Auto
t_dur_r_car     :           TIMER;
anz             {S7_server := 'alarm_archiv'; S7_a_type := 'alarm_8'} :DWORD;
// Anzahl Autos
// anz besitzt Systemattribute fuer Parameter

END_VAR
VAR_OUTPUT

g_car           :           BOOL            :=      FALSE;  // GRUEN fuer Autos

END_VAR
VAR
condition       :           BOOL    :=      FALSE;  // Vormerkung Rot fuer Autos
END_VAR

BEGIN
NETWORK
TITLE = Vormerkung Rot fuer Strassenverkehr
// Nach Einhalten eines Mindestzeitabstands wird durch
//den Signalwunsch am Fussgaengeruebergang die
//Vormerkung Rot fuer den Strassenverkehr gebildet.
                    U(;
                    U       #starter;       // Signalwunsch Fussgaengeruebergang und
                    U       #t_next_r_car;   // Zeit zwischen zwei Rotphasen abgelaufen
                    O       #condition;     // oder Vormerkung fuer Rot ("Selbsthaltung")
                    );
                    UN      #t_dur_y_car;   // und z.Zt. keine Rote Ampel
                    =       #condition;     // Vormerkung Rot
```

```
NETWORK
TITLE = Gruene Ampel fuer Strassenverkehr
                    UN      #condition;     // keine Vormerkung Rot Strassenverkehr
                    =       #g_car;         // GRUEN fuer Strassenverkehr
NETWORK
TITLE = Dauer fuer Gelbphase Autos
                    // weiteres Programm zur Realisierung
                    // der Ampelschaltung

END_FUNCTION_BLOCK

FUNCTION_BLOCK FB 10
VAR_INPUT
  para1 : bool;
  para2: bool;
end_var
begin
end_function_block

data_block db 10
fb10
begin
end_data_block

data_block db 6
fb6
begin
end_data_block
```

11.7.5 Beispiele für DBs in AWL-Quellen

Datenbaustein:

```
DATA_BLOCK DB 10
TITLE = DB Beispiel 10
STRUCT
                    aa : BOOL;      // Variable aa vom Typ BOOL
                    bb : INT; // Variable bb vom Typ INT
                    cc : WORD;
END_STRUCT;
BEGIN               // Zuweisung von Anfangswerten
                    aa := TRUE;
                    bb := 1500;
END_DATA_BLOCK
```

DB mit zugeordnetem anwenderdefiniertem Datentyp:

```
DATA_BLOCK DB 20
TITLE = DB (UDT) Beispiel
UDT 20                    // Angabe des zugeordneten UDTs
BEGIN
                start := TRUE;          // Zuweisung von Anfangswerten
                soll := 10;
END_DATA_BLOCK
```

Hinweis
Der verwendete UDT muß in der Quelle vor dem Datenbaustein stehen.

DB mit zugeordnetem Funktionsbaustein:

```
DATA_BLOCK DB 30
TITLE = DB (FB) Beispiel
FB 30           // Angabe des zugeordneten FBs
BEGIN
                start := TRUE;      // Zuweisung von Anfangswerten
                soll := 10;
END_DATA_BLOCK
```

Hinweis
Der zugeordnete FB muß in der Quelle vor dem Datenbaustein stehen.

11.7.6 Beispiel für UDTs in AWL-Quellen

```
TYPE  UDT 20
STRUCT
                start : BOOL;           //Variable vom Typ BOOL
                soll : INT;             //Variable vom Typ INT
                wert : WORD;            //Variable vom Typ WORD
END_STRUCT;
END_TYPE
```

12 Anzeigen von Referenzdaten

12.1 Übersicht der möglichen Referenzdaten

12.1.1 Übersicht der möglichen Referenzdaten

Sie können Referenzdaten erzeugen und auswerten, um sich das Testen und Ändern Ihres Anwenderprogramms zu erleichtern. Referenzdaten werden beispielsweise benutzt als

- Übersicht über ein gesamtes Anwenderprogramm,
- Grundlage für Änderungen und Tests,
- Ergänzung der Programmdokumentation.

Die folgende Tabelle gibt Auskunft darüber, welche Informationen Sie den einzelnen Ansichten entnehmen können:

Ansicht	Anwendung
Querverweisliste	Überblick über die Verwendung von Operanden der Speicherbereiche E, A, M, P, T, Z und DB-, FB-, FC-, SFB- und SFC-Aufrufen innerhalb des Anwenderprogramms. Mit dem Menübefehl **Ansicht > Querverweise für Operand** können Sie sich alle Querverweise einschließlich der überlappenden Zugriffe auf den markierten Operanden anzeigen lassen.
Belegungsplan Belegungsplan für Zeiten und Zähler (T/Z)	Der Überblick, welche Bits der Operanden der Speicherbereiche E, A, und M bzw. welche Zeiten und Zähler innerhalb des Anwenderprogramms bereits belegt sind, ist eine wichtige Grundlage zur Fehlersuche oder für Änderungen im Anwenderprogramm.
Programmstruktur	Aufrufhierarchie der Bausteine innerhalb eines Anwenderprogramms und Überblick über die verwendeten Bausteine und deren Abhängigkeiten.
Nicht verwendete Symbole	Überblick über alle Symbole, die in der Symboltabelle definiert sind, aber in den Teilen des Anwenderprogramms nicht verwendet werden, für die Referenzdaten vorliegen.
Operanden ohne Symbol	Überblick über alle Absolutadressen (Absolutoperanden und Bausteine), die in den Teilen des Anwenderprogramms verwendet werden und für die Referenzdaten vorliegen, für die aber in der Symboltabelle kein Symbol definiert ist.

Die Referenzdaten des ausgewählten Anwenderprogramms umfassen die in der Tabelle angegebenen Listen. Sie können sich mehrere Listen zu einem Anwenderprogramm oder zu unterschiedlichen Programmen erzeugen und anzeigen lassen.

Gleichzeitige Anzeige mehrerer Ansichten

Das Anzeigen von Listen in zusätzlichen Fenstern ermöglicht Ihnen zum Beispiel:

- eine Gegenüberstellung von gleichen Listen zu unterschiedlichen S7-Anwenderprogrammen.
- verschiedene Ansichten einer Liste, z.B. einer Querverweisliste, unterschiedlich optimiert, nebeneinander auf dem Bildschirm zu haben. Sie können sich z.B. in der einen Querverweisliste nur die Eingänge, in der anderen nur die Ausgänge eines S7-Anwenderprogramms anzeigen lassen.
- mehrere Listen zu einem S7-Anwenderprogramm gleichzeitig zu öffnen, z.B. Programmstruktur und Querverweisliste.

12.1.2 Querverweisliste

Die Querverweisliste bietet einen Überblick über die Verwendung von Operanden innerhalb des S7-Anwenderprogramms.

Mit der Anzeige der Querverweisliste erhalten Sie eine Liste der im S7-Anwenderprogramm verwendeten Operanden der Speicherbereiche Eingang (E), Ausgang (A), Merker (M), Zeiten (T), Zähler (Z), Funktionsbaustein (FB), Funktion (FC), Systemfunktionsbaustein (SFB), Systemfunktion (SFC), Peripherie (P) und Datenbaustein (DB), ihrer Adressen (absolute Adresse, Name) und ihrer Verwendung. Sie wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Querverweisliste gehört.

Jede Zeile im Fenster entspricht einem Querverweislisteneintrag. Eine Suchfunktion erleichtert Ihnen das gezielte Auffinden bestimmter Operanden und Symbole.

Die Querverweisliste ist die voreingestellte Ansicht beim Anzeigen von Referenzdaten. Sie können diese Voreinstellung ändern.

Aufbau

Ein Querverweislisteneintrag besteht aus folgenden Spalten:

Spalte	Inhalt/Bedeutung
Operand	absolute Adresse des Operanden
Symbol	Name des Operanden
Baustein	Angabe des Bausteins, in dem der Operand verwendet wird
Art	Angabe, ob es sich um einen lesenden (R) und/oder schreibenden (W) Zugriff auf den Operanden handelt
Sprache/Details	sprachabhängige Informationen, hängen von der Erstellungssprache des Bausteins ab

Die Spalten Symbol, Baustein, Art und Sprache/Details werden nur angezeigt, wenn die entsprechenden Eigenschaften für die Querverweisliste angewählt worden sind. Die Informationen Sprache und Details werden in einer gemeinsamen Spalte dargestellt und können nur als ganze Spalte an- oder abgewählt werden. Diese sprachabhängigen Informationen variieren je nach Erstellungssprache des Bausteins.

Die Spaltenbreite können Sie in der auf dem Bildschirm angezeigten Querverweisliste über Mausbedienung an ihre Anforderungen anpassen.

Sortierung

Voreingestellt in der Querverweisliste ist die Sortierung nach Speicherbereichen. Wenn Sie mit der Maus auf eine Spaltenüberschrift klicken, wird nach den Einträgen dieser Spalte sortiert.

Beispiel für den Aufbau der Querverweisliste

Operand	Symbol	Baustein	Art	Sprache	Details
E 1.0	Motor ein	OB 2	R	AWL	NW 2 Anw 33 /O
M1.2	MerkerBit	FC 2	R	KOP	NW 33
Z2	Zähler2	FB2		FUP	NW2

12.1.3 Programmstruktur

Die Programmstruktur beschreibt die Aufrufhierarchie der Bausteine innerhalb eines S7-Anwenderprogramms. Außerdem erhalten Sie einen Überblick über die verwendeten Bausteine, deren Abhängigkeiten und deren Lokaldatenbedarf.

Mit dem Menübefehl **Ansicht > Filtern** im Fenster "S7-Referenzdaten anzeigen" öffnen Sie einen Registerdialog. Im Register ”Programmstruktur” können Sie festlegen, wie die Programmstruktur dargestellt wird.

Sie haben die Wahl zwischen

- Baumdarstellung und
- Paarweise Aufrufbeziehung.

Sie können angeben, ob alle Bausteine dargestellt werden sollen oder ob die Hierarchie ab einem bestimmten Startbaustein angezeigt werden soll.

Symbole in der Programmstruktur

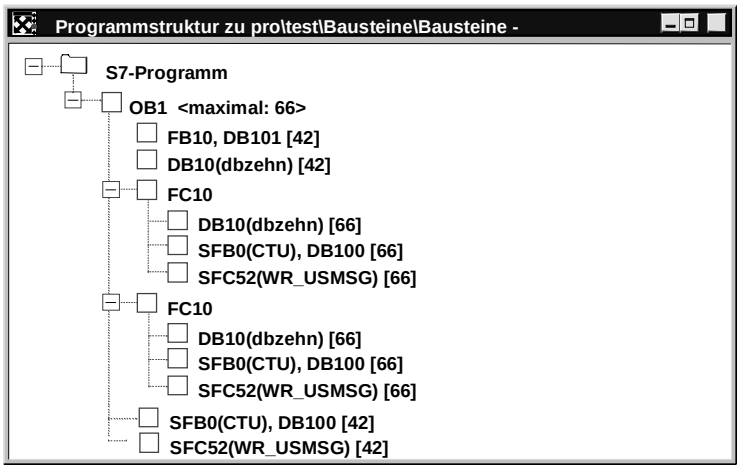
Symbol	Bedeutung
	Baustein regulär aufgerufen (CALL FB10)
	Baustein unbedingt aufgerufen (UC FB10)
	Baustein bedingt aufgerufen (CC FB10)
	Datenbaustein
	Rekursion
	Rekursion und bedingt aufgerufen
	Rekursion und unbedingt aufgerufen
	Baustein nicht aufgerufen

- Rekursionen im Aufruf werden erkannt und in der Baumstruktur grafisch gekennzeichnet.
- Rekursionen innerhalb der Aufrufhierarchie werden durch unterschiedliche Symbole dargestellt.
- Regulär aufgerufene Bausteine (CALL), bedingt aufgerufene Bausteine (CC) oder unbedingt aufgerufene Bausteine (UC) werden durch unterschiedliche Symbole gekennzeichnet.
- Nicht aufgerufene Bausteine werden am unteren Ende der Baumstruktur angezeigt und mit einem schwarzen Kreuz markiert. Die Aufrufstruktur eines nicht aufgerufenen Bausteins wird nicht weiter aufgeschlüsselt.

Anzeige als Baumstruktur

Es wird die gesamte Aufrufhierarchie, ausgehend von einem bestimmten Baustein, dargestellt.

Jede Programmstruktur besitzt genau einen Baustein als Wurzel. Dieser Baustein kann der OB1 sein oder ein beliebiger anderer Baustein, der vom Anwender als Startbaustein vorgegeben wird.



Soll die Programmstruktur für alle Organisationsbausteine (OB) erstellt werden und der OB1 ist nicht im S7-Anwenderprogramm, oder wurde ein Startbaustein angegeben, der nicht im Programm vorhanden ist, werden Sie automatisch zur Vorgabe eines anderen Bausteins für die Wurzel der Programmstruktur aufgefordert.

Die Anzeige von Mehrfachaufrufen von Bausteinen kann über Optionseinstellungen sowohl für die Baumstruktur als auch für die "paarweise Aufrufbeziehung" ausgeschaltet werden.

Anzeige des maximalen Lokaldatenbedarfs in der Baumstruktur

Um Ihnen einen schnellen Überblick über den Lokaldatenbedarf der OBs im angezeigten Anwenderprogramm zu geben, können Sie sich in der Baumdarstellung folgendes anzeigen lassen:

- den Maximalbedarf der Lokaldaten pro OB
- den Lokaldatenbedarf pro Pfad

Die Anzeige läßt sich im Register ”Programmstruktur” ein- und ausschalten.

Um dann den Lokaldatenbedarf eines markierten Bausteins zu erhalten, klicken Sie auf die rechte Maustaste und wählen aus dem Kontextmenü den Befehl **Bausteininformation** aus.

Falls synchrone Fehler-OBs (OB 121, OB 122) vorhanden sind, wird bei Anzeige des maximalen Lokaldatenbedarfs nach dem Zahlenwert ein Pluszeichen und danach der zusätzliche Bedarf für die synchronen Fehler-OBs angegeben.

Anzeige als paarweise Aufrufbeziehung

Es wird der aufrufende und der aufgerufene Baustein dargestellt. Diese paarweise Aufrufbeziehung wird für jeden im S7-Anwenderprogramm liegenden Baustein angegeben.

Anzeige gelöschter Bausteine

Die Zeilen zu gelöschten Bausteinen werden rot hervorgehoben und hinter den Bausteinen wird die Zeichenkette "?????" angezeigt.

12.1.4 Belegungsplan für Eingänge, Ausgänge und Merker (E/A/M)

Belegungspläne zeigen Ihnen, welche Operanden innerhalb des Anwenderprogramms bereits belegt sind. Diese Anzeige ist eine wichtige Grundlage für Fehlersuche oder Änderungen im Anwenderprogramm.

Mit der Anzeige des Belegungsplans E/A/M erhalten Sie eine Übersicht, welches Bit in welchem Byte der Speicherbereiche Eingang (E), Ausgang (A), und Merker (M) verwendet wird. Der Belegungsplan wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des S7-Anwenderprogramms an, zu dem der Belegungsplan gehört.

Jede Zeile beinhaltet ein Byte des Speicherbereichs, in dem die acht Bits je nach Zugriff gekennzeichnet werden. Zusätzlich wird noch angegeben, ob der Zugriff durch einen Byte-, Wort- oder Doppelwortzugriff erfolgt.

Kennzeichen im Belegungsplan E/A/M:

. der Operand ist nicht angesprochen und somit noch nicht belegt

o der Operand ist direkt verwendet

x der Operand ist indirekt bearbeitet (Byte-, Wort- oder Doppelwortzugriff)

Spalten im Belegungsplan E/A/M

Spalte	Inhalt/Bedeutung
7 6 5 4 3 2 1 0	Bitnummer des entsprechenden Bytes
B	das Byte ist durch einen Bytezugriff belegt
W	das Byte ist durch einen Wortzugriff belegt
D	das Byte ist durch einen Doppelwortzugriff belegt

Beispiel eines Belegungsplans (E/A/M)

Folgendes Beispiel zeigt den typischen Aufbau eines Belegungsplans für Eingänge, Ausgänge und Merker (E/A/M).

Operand	7	6	5	4	3	2	1	0		B	W	D
AB0	O	X	X	O	X	X	X	X		O	.	.
AB1	.	O	.	.	O	.	O	.		.	.	.
EB0	O	O	O	.	O	.	O	.		.	.	.
EB1	.	.	.	.	.	.	.	.		.	.	.
MB0	X	X	X	X	X	X	X	X		.	O	.
MB1	X	X	X	X	X	X	O	X		.	.	.

In der ersten Zeile ist die Belegung vom Ausgangsbyte AB 0 aufgeführt. Es wird auf den Operanden AB 0 byte-weise zugegriffen. Es erfolgt aber gleichzeitig ein Bitzugriff auf die Ausgangsbits A 0.4 und A 0.7. Daher steht in den Spalten "4" und "7" ein "O". In den Spalten "0", "1", "2", "3", "5", und "6" steht jeweils ein "X" als Kennzeichen für den Bytezugriff. In der Spalte "B" steht ein "O", da auf den Operanden AB 0 ein Bytezugriff erfolgt.

12.1.5 Belegungsplan für Zeiten und Zähler (T/Z)

Belegungspläne zeigen Ihnen, welche Operanden innerhalb des Anwenderprogramms bereits belegt sind. Diese Anzeige ist eine wichtige Grundlage für Fehlersuche oder Änderungen im Anwenderprogramm.

Mit der Anzeige des Belegungsplans T/Z erhalten Sie eine Übersicht, welche Zeiten (T) und Zähler (Z) verwendet sind.

Der Belegungsplan wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des S7-Anwenderprogramms an, zu dem der Belegungsplan gehört. In jeder Zeile werden 10 Zeiten bzw. Zähler dargestellt.

Kennzeichen im Belegungsplan T/Z:

. nicht verwendet

x verwendet

Beispiel eines Belegungsplans T/Z

	0	1	2	3	4	5	6	7	8	9
T 00-09	.	X	.	.	.		X	.	.	.
T 10-19	.	.	X	.	.	.	.	X	.	X
T 20-29	.	.	.	.	X	.	.	.	.	.
Z 00-09	.	.	X	.	.	.	.	X	.	.
Z 10-19	.	.	.	.	.	.	.	.	.	X
Z 20-29	.	.	.	.	.	.	.	.	.	.
Z 30-39	.	.	.	.	X	.	.	.	.	.

In diesem Beispiel sind die Zeiten T1, T6, T12, T17, T19, T24 und die Zähler Z2, Z7, Z19, Z34 belegt.

Diese Listen sind alphabetisch sortiert. Die Einträge können Sie durch Anklicken der Spaltenüberschrift anordnen.

12.1.6 Nicht verwendete Symbole

Sie erhalten einen Überblick über alle Symbole mit folgender Eigenschaft:

- Die Symbole sind in der Symboltabelle definiert.
- Die Symbole werden aber innerhalb der Anwenderprogrammteile, für die Referenzdaten vorliegen, nicht verwendet.

Sie wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Liste gehört.

Jede angezeigte Zeile im Fenster entspricht einem Listeneintrag. Eine Zeile besteht aus Operand, Symbol, Datentyp und Kommentar.

Spalte	Inhalt/Bedeutung
Symbol	Name
Operand	absolute Adresse des Operanden
Datentyp	Datentyp des Operanden
Operandenkommentar	Kommentar des Operanden aus der Symboltabelle

Beispiel einer Liste nicht verwendeter Symbole

Symbol	Operand	Datentyp	Operandenkommentar
MS1	E103.6	BOOL	Motorschutzschalter 1
MS2	E120.5	BOOL	Motorschutzschalter 2
MS3	E121.3	BOOL	Motorschutzschalter 3

Die Einträge können Sie durch Anklicken der Spaltenüberschrift anordnen.

12.1.7 Operanden ohne Symbol

Mit der Anzeige der Liste der Operanden ohne Symbol erhalten Sie eine Liste der Elemente, die im S7-Anwenderprogramm verwendet werden, in der Symboltabelle aber nicht definiert sind. Sie wird in einem Arbeitsfenster angezeigt. Die Titelzeile des Arbeitsfensters zeigt den Namen des Anwenderprogramms an, zu dem die Liste gehört. Eine Zeile besteht aus dem Operanden und der Angabe, wie oft er verwendet worden ist.

Beispiel:

Operand	Anzahl
A 2.5	4
E 23.6	3
M 34.1	20

Die Einträge sind nach Operanden geordnet.

12.1.8 Anzeige sprachabhängiger Informationen zu KOP, FUP, AWL

Die sprachabhängigen Informationen zu KOP, FUP und AWL werden in der Querverweisliste und in der Programmstruktur angezeigt. Sie bestehen aus der Bausteinsprache und Details.

In der Ansicht "Programmstruktur" werden die sprachabhängigen Informationen über den Menübefehl **Ansicht > Bausteininformationen** oder über die rechte Maustaste aufgerufen. Es ist davon abhängig, ob bei den Filtereinstellungen im Register "Programmstruktur" die Darstellung "paarweise Aufrufbeziehung" oder "Baumdarstellung" gewählt wurde.

In der Ansicht "Querverweise" können die sprachabhängigen Informationen über **Ansicht > Filtern** ein- oder ausgeblendet werden.

- Aktivieren Sie das Kontrollkästchen "Bausteinsprache und Details" im Register "Querverweise" des Dialogfeldes "Filtern", um die sprachabhängigen Informationen einzublenden.

Die sprachabhängigen Informationen variieren je nach Erstellungssprache des Bausteins und werden mit Kürzeln dargestellt.

Sprache	Netzwerk	Anweisung	Operation
AWL	NW	Anw	/
KOP	NW		
FUP	NW		

NW, Anw geben an, in welchem Netzwerk, in welcher Anweisung der Operand benutzt (Querverweise) bzw. der Baustein aufgerufen wird (Programmstruktur).

Anzeige sprachabhängiger Informationen zu optionalen Programmiersprachen

Die Hilfe zu sprachabhängigen Informationen ist aufrufbar, wenn das entsprechende Optionspaket installiert ist.

12.2 Arbeiten mit Referenzdaten

12.2.1 Möglichkeiten zum Anzeigen von Referenzdaten

Zum Anzeigen von Referenzdaten bestehen folgende Möglichkeiten:

Anzeigen aus dem SIMATIC Manager:

1. Selektieren Sie im Projektfenster in der Projektansicht offline den Ordner "Bausteine".
2. Wählen Sie den Menübefehl **Extras > Referenzdaten > Anzeigen**.

Anzeigen aus dem Editor-Fenster:

1. Öffnen Sie einen Baustein im Ordner ”Bausteine”.
2. Wählen Sie im Fenster des entsprechenden Spracheditors den Menübefehl **Extras** > **Referenzdaten**.

Der Dialog "Einstellungen" wird eingeblendet. Hier können Sie die Ansicht auswählen, die zuerst angezeigt werden soll. Es ist die Ansicht voreingestellt, die in der Applikation zum Anzeigen von Referenzdaten zuletzt geschlossen wurde. Der Dialog kann für künftige Aufrufe unterdrückt werden.

Wenn die Referenzdaten unvollständig sind, wird ein Dialogfeld angezeigt, von dem aus Sie eine Aktualisierung der Referenzdaten anstoßen können .

Anzeigen direkt aus dem kompilierten Baustein

Die Referenzdaten zu einem kompilierten Baustein können Sie direkt aus dem Spracheditor heraus anzeigen und sich damit einen aktuellen Überblick über Ihr Anwenderprogramm verschaffen.

12.2.2 Anzeigen von Listen in zusätzlichen Arbeitsfenstern

Mit dem Menübefehl **Fenster > Neues Fenster** können Sie weitere Arbeitsfenster öffnen und sich darin andere Sichten von bereits angezeigten Referenzdaten (z.B. Liste der nicht verwendeten Symbole) anzeigen lassen.

Ein Arbeitsfenster für noch nicht angezeigte Referenzdaten öffnen Sie mit dem Menübefehl **Referenzdaten > Öffnen**.

Sie können auf eine andere Ansicht der Referenzdaten umschalten, indem Sie den Menübefehl **Ansicht** oder die zugehörige Schaltfläche in der Funktionsleiste wählen:

Referenzdaten-Ansicht	**Menübefehl zur Anzeige dieser Referenzdaten-Ansicht**
Operanden ohne Symbol	Ansicht > Operanden ohne Symbol
Nicht verwendete Symbole	Ansicht > Nicht verwendete Symbole
Belegung E/A/M	Ansicht > Belegung > Eingänge, Ausgänge und Merker
Belegung T/Z	Ansicht > Belegung > Zeiten und Zähler
Programmstruktur	Ansicht > Programmstruktur
Querverweise	Ansicht > Querverweise

12.2.3 Erzeugen und Anzeigen von Referenzdaten

Erzeugen von Referenzdaten:

1. Markieren Sie im SIMATIC Manager den Bausteinordner, für den Referenzdaten generiert werden sollen.
2. Wählen Sie im SIMATIC Manager den Menübefehl **Extras > Referenzdaten > Generieren**.

Vor dem Generieren von Referenzdaten wird zunächst geprüft, ob Referenzdaten vorliegen und ob eventuell vorliegende Referenzdaten aktuell sind.

- Falls keine Referenzdaten vorliegen, werden sie generiert.
- Falls nicht aktuelle Referenzdaten vorliegen, können Sie in einem Dialogfeld entscheiden, ob die Referenzdaten aktualisiert oder komplett neu generiert werden sollen.

Anzeigen von Referenzdaten:

Mit dem Menübefehl **Extras > Referenzdaten > Anzeigen** können Sie sich die Referenzdaten anzeigen lassen.

Vor dem Anzeigen von Referenzdaten wird zunächst geprüft, ob Referenzdaten vorliegen und ob eventuell vorliegende Referenzdaten aktuell sind.

- Falls keine Referenzdaten vorliegen, werden sie erzeugt.
- Falls unvollständige Referenzdaten vorliegen, wird ein Dialogfeld mit einem Hinweis auf die Inkonsistenz der Referenzdaten angezeigt. Sie können dann entscheiden, ob und in welchem Umfang die Referenzdaten aktualisiert werden sollen.

Dabei haben Sie folgende Auswahlmöglichkeiten:

Wahl	Bedeutung
nur für geänderte Bausteine	Die Referenzdaten zu geänderten und neuen Bausteine werden aktualisiert; Information zu gelöschten Bausteinen wird aus den Referenzdaten entfernt.
für alle Bausteine	Die Referenzdaten für alle Bausteine werden komplett neu erstellt.
nicht aktualisieren	Die Referenzdaten werden nicht aktualisiert.

Zur Aktualisierung der Referenzdaten werden die Bausteine neu übersetzt. Dabei wird für jeden Baustein der passende Compiler aufgerufen. Mit dem Menübefehl **Ansicht > Aktualisieren** können Sie bereits angezeigte Referenzdaten im aktiven Fenster aktualisieren.

12.2.4 Schnelles Positionieren auf Verwendungsstellen im Programm

Sie können Referenzdaten benutzen, um beim Programmieren auf die Verwendungsstellen eines Operanden zu positionieren. Dazu müssen aktuelle Referenzdaten vorliegen. Der Aufruf der Applikation zum Anzeigen von Referenzdaten ist nicht erforderlich.

Prinzipielle Vorgehensweise

1. Wählen Sie im SIMATIC Manager den Menübefehl **Extras > Referenzdaten > Generieren**, um aktuelle Referenzdaten zu erzeugen. Dieser Schritt ist nur erforderlich, wenn Referenzdaten nicht vorliegen oder veraltet sind.
2. Markieren Sie in einem geöffneten Baustein den Operanden.
3. Wählen Sie den Menübefehl **Bearbeiten > Gehe zu > Verwendungsstelle**. Nun wird ein Dialogfeld angezeigt, das eine Liste mit den Verwendungsstellen des Operanden im Programm enthält.
4. Wählen Sie die Option "Überlappender Zugriff auf Speicherbereiche", wenn zusätzlich die Verwendungsstellen der Operanden angezeigt werden sollen, deren Adressen oder Adreßbereich die Adresse bzw. den Adreßbereich des aufgerufenen Operanden überlappen. Die Tabelle wird um die Spalte "Operand"erweitert.
5. Markieren Sie eine Verwendungsstelle in der Liste und klicken Sie auf die Schaltfläche "Gehe zu".

Wenn beim Aufruf des Dialogs die Referenzdaten nicht aktuell sind, werden Sie mit einer Meldung darauf hingewiesen. Sie können dann die Referenzdaten aktualisieren lassen.

Liste der Verwendungsstellen

Die Liste der Verwendungsstellen im Dialogfeld enthält folgende Angaben:

- Baustein, in dem der Operand verwendet wird
- Bausteinsymbol, falls vorhanden
- Details, d.h. von der Erstellsprache des Bausteins/der Quelle (SCL) abhängige Information zur Verwendungsstelle und ggf. der Operation
- sprachabhängige Informationen
- Zugriffsart auf den Operanden: Lesend (R), schreibend (W), lesend und schreibend (RW), nicht ermittelbar (?).
- Bausteinsprache

Sie können die Anzeige der Verwendungsstellen filtern, und sich beispielsweise nur die schreibenden Zugriffe auf einen Operanden anzeigen lassen. Ausführliche Information zu den Eingabemöglichkeiten und Anzeigen finden Sie in der Online-Hilfe zu diesem Dialogfeld.

Hinweis

Referenzdaten sind nur offline vorhanden. Diese Funktion arbeitet daher stets mit den Querverweisen der Bausteine offline, auch wenn Sie die Funktion in einem Baustein online aufrufen.

12.2.5 Beispiel zum Arbeiten mit Verwendungsstellen

Sie möchten feststellen, an welchen Stellen der Ausgang A1.0 (direkt/indirekt) gesetzt wird. Als Beispiel dient der folgende AWL-Code im Baustein OB1:

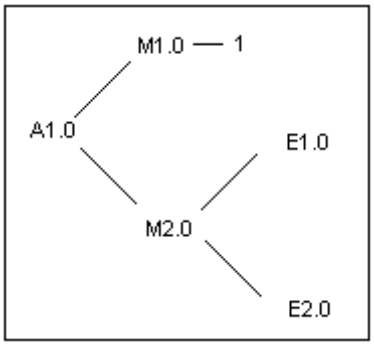
```
Netzwerk1: ...........
U A 1.0  // in diesem Beispiel
= A 1.1  // nicht relevant

Netzwerk 2:
U M1.0
U M2.0
= A 1.0  // Zuweisung

Netzwerk3:
//nur Kommentarzeile
SET
= M1.0  // Zuweisung
```

```
Netzwerk 4:
U E 1.0
U E 2.0
= M2.0  // Zuweisung
```

Für A1.0 ergibt sich daraus der folgende Zuweisungsbaum:



Dann gehen Sie so vor:

1. Im KOP/AWL/FUP-Editor im OB 1 auf A1.0 (NW 1, Anw 1) positionieren.
2. Über **Bearbeiten > Gehe zu > Verwendungsstelle** oder mit der rechten Maustaste "Gehe zur Verwendungsstelle" aufrufen.
   Im Dialogfeld werden unter anderem alle Zuweisungen auf A1.0 angezeigt:
   OB1 Cycle Execution NW 2 Anw 3 /= W AWL
   OB1 Cycle Execution NW 1 Anw 1 /U R AWL
3. Mittels "Gehe zu" aus dem Dialogfeld zu "NW 2 Anw 3" in den Editor springen:
   `Netzwerk 2:`
   U M1.0
   U M2.0
   = A 1.0
4. Sowohl die Zuweisungen an M1.0 als auch an M2.0 müssen nun kontrolliert werden. Also im KOP/AWL/FUP-Editor zuerst auf M1.0 positionieren.
5. Über **Bearbeiten > Gehe zu > Verwendungsstelle**" oder mit rechter Maustaste "Gehe zur Verwendungsstelle" aufrufen.Im Dialogfeld werden unter anderem alle Zuweisungen auf `M1.0` angezeigt:
   OB1 Cycle Execution NW 3 Anw 2 /= W AWL
   OB1 Cycle Execution NW 2 Anw 1 /U R AWL
6. Mittels "Gehe zu" zu "NW 3 Anw 2" in den KOP/AWL/FUP-Editor springen.

7. Im KOP/AWL/FUP-Editor in Netzwerk 3 wird festgestellt, daß die Belegung von M1.0 uninteressant ist (da immer TRUE) und stattdessen die Belegung von M2.0 untersucht werden müßte.

   Im STEP 7-Versionen vor V5 mußte jetzt die gesamte Kette von Zuweisungen nochmal ganz von vorne durchlaufen werden. Die Schaltflächen ">>" und "<<" vereinfachen das weitere Vorgehen:

8. Entweder das noch geöffnete Dialogfeld "Gehe zu Verwendungsstelle" nach vorne holen oder aus der aktuellen Position im KOP/AWL/FUP-Editor "Gehe zur Verwendungsstelle" aufrufen.
9. Die Schaltfläche "<<" einmal oder zweimal betätigen, bis alle Verwendungsstellen von A 1.0 angezeigt werden, wobei die letzte Aussprungstelle "NW 2 Anw 3" markiert ist.
10. Mittels "Gehe zu" (wie in Punkt 3) aus dem Verwendungsstellendialog zu "`NW 2 Anw 3`" in den Editor zu springen:
    `Netzwerk 2:`
    U M1.0
    U M2.0
    = A 1.0
11. In Punkt 4ff. wurde die Zuweisung an `M1.0` kontrolliert. Jetzt müssen alle (direkten/indirekten) Zuweisungen auf `M2.0` kontrolliert werden. Also im Editor auf M2.0 positionieren und "Gehe zur Verwendungsstelle" aufrufen: Es werden u.a. alle Zuweisungen auf `M2.0` angezeigt:
    OB1 Cycle Execution NW 4 Anw 3 /= W AWL
    OB1 Cycle Execution NW 2 Anw 2 /U R AWL
12. Mittels "Gehe zu" zu "NW 4 Anw 3" in den KOP/AWL/FUP-Editor springen:
    `Netzwerk 4:`
    U E 1.0
    U E 2.0
    = M2.0
13. Jetzt müssen die Zuweisungen an E1.0 und E2.0 überprueft werden. Dies wird in diesem Beispiel nicht mehr erläutert, da die weitere Vorgehensweise sich von der bisherigen (Punkt 4ff.) nicht unterscheidet

Durch abwechselndes Springen zwischen KOP/AWL/FUP-Editor und dem Verwendungsstellen-Dialog können sie so die relevanten Stellen ihres Programms ermitteln und untersuchen.

13 Bausteinkonsistenz prüfen und Zeitstempel als Bausteineigenschaft

13.1 Bausteinkonsistenz prüfen

Einführung

Müssen während oder nach der Programmerstellung die Schnittstellen bzw. der Code einzelner Bausteine angepaßt oder ergänzt werden, so kann dies zu Zeitstempelkonflikten führen. Zeitstempelkonflikte können wiederum zu Bausteininkonsistenzen zwischen aufrufenden und aufgerufenen Bausteinen bzw. Referenzbausteinen und damit zu einem hohen Korrekturaufwand führen.

Die Funktion "Bausteinkonsistenz prüfen" nimmt Ihnen einen großen Teil dieser Korrekturarbeit ab. Die Funktion "Bausteinkonsistenz prüfen" bereinigt sebständig einen großen Teil aller Zeitstempelkonflikte und Bausteininkonsistenzen. Bei Bausteinen, deren Bausteininkonsistenzen nicht automatisch bereinigt werden konnten, führt Sie die Funktion an die zu ändernden Positionen im dazugehörigen Editor. Dort nehmen Sie die notwendigen Änderungen vor. Schritt für Schritt werden alle Bausteininkonsistenzen bereinigt und die Bausteine übersetzt.

Voraussetzungen

Die Bausteinkonsistenz-Prüfung ist nur bei Projekten möglich, die ab STEP 7 V5.0, Servicepack 3 erstellt wurden. Bei älteren Projekten müssen Sie daher beim Starten der Bausteinkonsistenz-Prüfung zunächst alles übersetzen (Menübefehl **Programm > Alles übersetzen**).

Bei Bausteinen, die mit einem Optionspaket erstellt wurden, muß das Optionspaket für eine Konsistenzprüfung installiert sein.

Starten der Bausteinkonsistenz-Prüfung

Mit dem Start der Bausteinkonsistenzprüfung werden die Zeitstempel der Bausteinschnittstellen geprüft und Bausteine, bei denen es zu Bausteininkonsistenzen führen kann, werden in der Baumansicht (Abhängigkeitsbaum/Referenzbaum) markiert.

1. Wählen Sie den Menübefehl **Programm > Übersetzen**.
   STEP 7 erkennt automatisch die Erstellsprache für die betreffenden Bausteine und ruft den entsprechenden Editor auf. Soweit wie möglich, werden die Zeitstempelkonflikte und Bausteininkonsistenzen automatisch bereinigt und die Bausteine übersetzt. Konnte der Zeitstempelkonflikt bzw. die Inkonsistenz in einem Baustein nicht automatisch bereinigt werden, so erfolgt eine Fehlermeldung im Ausgabefenster (weitere Vorgehensweise siehe Schritt 2). Dieser Ablauf wird automatisch für alle Bausteine der Baumansicht wiederholt.
2. Konnten mit dem Übersetzungslauf nicht alle Bausteininkonsistenzen automatisch bereinigt werden, so werden die betreffenden Bausteine im Ausgabefenster als Fehlermeldung markiert. Positionieren Sie mit der Maus den entsprechenden Fehlereintrag und wählen Sie mit der rechten Maustaste im Kontextmenübefehl Fehler anzeigen. Der entsprechende Editor wird geöffnet und an die zu ändernden Positionen gesprungen. Beheben Sie alle Bausteininkonsistenzen und schließen und speichern Sie den Baustein. Wiederholen Sie den Ablauf für alle als Fehler markierten Bausteine.
3. Starten Sie Schritt 1 und 2 erneut. Wiederholen Sie diesen Vorgang solange, bis im Ausgabefenster keine Fehler mehr angezeigt werden.

13.2 Zeitstempel als Bausteineigenschaft und Zeitstempelkonflikte

Bausteine enthalten einen Code-Zeitstempel und einen Schnittstellen-Zeitstempel. Diese Zeitstempel werden im Dialogfeld zu den Baustein-Eigenschaften angezeigt. Anhand von Zeitstempeln wird die Konsistenz von STEP 7-Programmen überwacht.

STEP 7 zeigt einen Zeitstempelkonflikt an, wenn beim Vergleich von Zeitstempeln ein Regelverstoß erkannt wird. Folgende Regelverstöße können auftreten:

- Ein aufgerufener Baustein ist jünger als der aufrufende Baustein (CALL).
- Ein referenzierter Baustein ist jünger als der Baustein, der ihn verwendet.
- Beispiele zum zweiten Punkt:
- Ein UDT ist jünger als der Baustein, der ihn verwendet, z.B.ein DB oder ein anderer UDT, oder ein FC, FB, OB, der den UDT in der Variablendeklarationstabelle verwendet.
- Ein FB ist jünger als der zugehörige Instanz-DB.
- In FB1 ist ein FB2 als Multiinstanz definiert und FB2 ist jünger als FB1.

Hinweis

Auch wenn die Relation der Schnittstellen-Zeitstempel korrekt ist, können Inkonsistenzen vorkommen:

- Die Definition der Schnittstelle des referenzierten Bausteins stimmt nicht mit der an seiner Verwendungsstelle verwendeten Schnittstelle überein.

Solche Inkonsistenzen werden Schnittstellenkonflikte genannt. Sie können z.B. durch Kopieren von Bausteinen verschiedener Programme entstehen oder durch den Übersetzungslauf einer ASCII-Quelle, bei dem nur eine Teilmenge der Bausteine eines Gesamtprogramms erzeugt wird.

13.3 Zeitstempel in Codebausteinen

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt des Bausteins eingetragen. Der Zeitstempel wird aktualisiert

- bei Änderung des Programmcodes
- bei Änderungen der Schnittstellenbeschreibung
- bei Änderung des Kommentars
- bei der Ersterstellung und der Übersetzung einer ASCII-Quelle
- bei Änderungen der Bausteineigenschaften (Dialog: Eigenschaften)

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert bei

- Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter)
- der Ersterstellung und der Übersetzung einer ASCII-Quelle, falls sich die Schnittstelle strukturell ändert.

Der Zeitstempel wird nicht aktualisiert bei:

- Änderung von Symbolen
- Änderung von Kommentaren in der Variablendeklaration
- Änderungen im TEMP-Bereich.

Regel für Bausteinaufrufe

- Der Schnittstellen-Zeitstempel des aufgerufenen Bausteins muß älter sein als der Code-Zeitstempel des aufrufenden Bausteins.
- Ändern Sie die Schnittstelle eines Bausteins nur, wenn kein Baustein geöffnet ist, der diesen aufruft. Denn wenn Sie die aufrufenden Bausteine später als den geänderten speichern, so erkennen sie diese Inkonsistenz nicht am Zeitstempel.

Vorgehen bei Zeitstempelkonflikt

Ein Zeitstempelkonflikt wird beim Öffnen des Aufrufers angezeigt. Nach Änderung einer FC- oder FB-Schnittstelle werden alle Aufrufe dieses Bausteins in rufenden Bausteinen in expandierter Form dargestellt.

Wird die Schnittstelle eines Bausteins geändert, so müssen alle Bausteine angepaßt werden, die diesen Baustein aufrufen.

Nach Änderung einer FB-Schnittstelle müssen die vorhandenen Multiinstanzdefinitionen und Instanz-Datenbausteine aktualisiert werden.

13.4 Zeitstempel bei globalen Datenbausteinen

Code-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei der Ersterstellung,
- bei der Übersetzung einer ASCII-Quelle,
- bei Änderungen in der Deklarationssicht oder in der Datensicht des Bausteins.

Schnittstellen-Zeitstempel

Der Zeitstempel wird aktualisiert

- bei Änderungen der Schnittstellenbeschreibung in der Deklarationssicht (Änderung von Datentypen oder Initialwerten, neue Parameter)

13.5 Zeitstempel bei Instanz-Datenbausteinen

Ein Instanz-Datenbaustein speichert die Formalparameter und statischen Daten von Funktionsbausteinen.

Code-Zeitstempel

Hier wird der Erstellungszeitpunkt der Instanz-Datenbausteine eingetragen. Der Zeitstempel wird aktualisiert, wenn Sie Aktualwerten in der Datensicht des Instanz-Datenbausteins eintragen. Änderungen der Struktur eines Instanz-Datenbausteins durch den Benutzer sind nicht möglich; da die Struktur vom zugehörigen Funktionsbaustein (FB) oder Systemfunktionsbaustein (SFB) abgeleitet ist.

Schnittstellen-Zeitstempel

Beim Erzeugen eines Instanz-Datenbausteins wird der Schnittstellen-Zeitstempel des zugehörigen FB oder SFB eingetragen.

Regeln für konfliktfreies Öffnen

Die Schnittstellen-Zeitstempel des FB/SFB und des zugeordneten Instanz-Datenbausteins müssen übereinstimmen.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie die Schnittstelle eines FB ändern, wird der Schnittstellen-Zeitstempel des FB aktualisiert. Beim Öffnen eines zugehörigen Instanz-Datenbausteins wird ein Zeitstempelkonflikt angezeigt, da die Zeitstempel des Instanz-Datenbausteins und des FB nicht mehr übereinstimmen. Im Deklarationsteil des DB wird die Schnittstelle mit vom Compiler generierten Symbolen (Pseudosymbolik) dargestellt. Der Instanz-Datenbaustein ist nur noch zur Ansicht nutzbar.

Um solche Zeitstempelkonflikt zu beheben, müssen Sie den zu einem geänderten FB gehörigen Instanz-DB neu erzeugen.

13.6 Zeitstempel bei UDTs und von UDTs abgeleiteten DBs

Anwenderdefinierte Datentypen (UDT) können z.B. zur Erzeugung mehrerer Datenbausteine mit der gleichen Struktur genutzt werden.

Code-Zeitstempel

Der Code-Zeitstempel wird aktualisiert bei jeder Änderung.

Schnittstellen-Zeitstempel

Der Schnittstellen-Zeitstempel wird bei Änderungen der Schnittstellenbeschreibung (Änderung von Datentypen oder Initialwerten, neue Parameter) aktualisiert.

Der Schnittstellen-Zeitstempel eines UDT wird auch bei Übersetzung der ASCII-Quelle aktualisiert.

Regeln für konfliktfreies Öffnen

- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muß älter sein als der Schnittstellen-Zeitstempel in Codebausteinen, in denen dieser Datentyp verwendet wird.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muß identisch sein mit dem Zeitstempel eines aus einem UDT abgeleiteten DB.
- Der Schnittstellen-Zeitstempel des anwenderdefinierten Datentyps muß jünger sein als der Zeitstempel eines unterlagerten UDT.

Vorgehen bei Zeitstempelkonflikt

Wenn Sie eine UDT-Definition ändern, die in einem DB, FC, FB oder einer anderen UDT-Definition verwendet wird, zeigt STEP 7 beim Öffnen eines solchen Bausteins ein Zeitstempelkonflikt an.

Die UDT-Komponente wird als Struktur aufgefächert dargestellt. Alle Variablennamen werden durch vom System vorgegebene Werte überschrieben.

14 Projektieren von Meldungen

14.1 Grundlagen des Meldekonzeptes

14.1.1 Grundlagen des Meldekonzeptes

Meldungen ermöglichen Ihnen, Fehler bei der Prozeßbearbeitung in den Automatisierungssystemen schnell zu erkennen, genau zu lokalisieren und zu beheben. Stillstandszeiten einer Anlage können so wesentlich verkürzt werden.

Bevor Meldungen ausgegeben werden können, müssen sie projektiert werden.

Mit STEP 7 können Sie ereignisabhängige Meldungen mit zugeordneten Meldungstexten und Meldungsattributen anlegen, bearbeiten, übersetzen und auf Anzeigegeräten ausgeben.

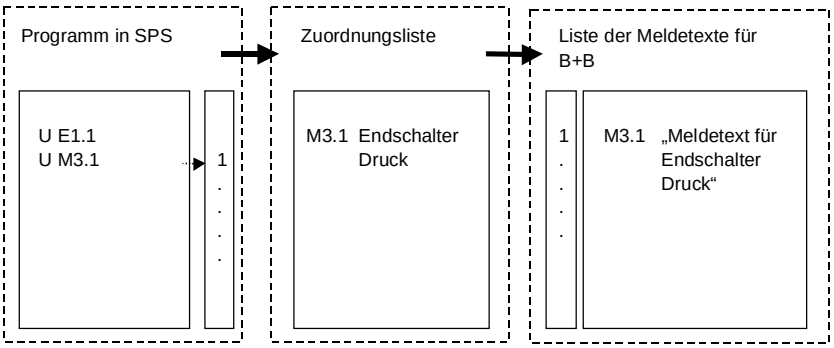
14.1.2 Welche Meldeverfahren gibt es?

Für das Erzeugen von Meldungen gibt es unterschiedliche Verfahren.

Bitmeldeverfahren

Das Bitmeldeverfahren erfordert 3 Arbeitsschritte durch den Programmierer:

- Er erstellt am PG das Anwenderprogramm und setzt das gewünschte Bit.
- Er erstellt mit einem beliebigen Texteditor eine Zuordnungsliste, in der dem Meldebit ein Meldetext zugewiesen wird (z. B. M 3.1 = Endschalter Druck).
- Er erstellt am Bediensystem auf der Basis der Zuordnungsliste die Liste der Meldetexte.

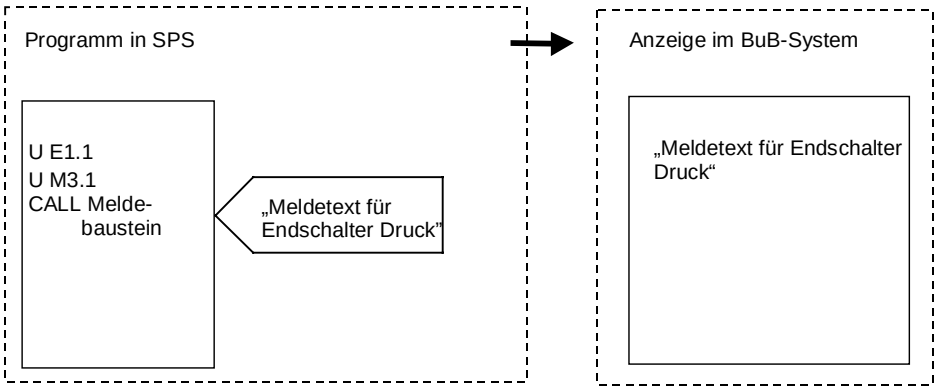


Das Bedien- und Beobachtungssystem fragt zyklisch das Automatisierungssystem ab, ob sich das Meldebit geändert hat oder nicht. Meldet das AS eine Änderung zurück, wird die entsprechende Meldung ausgegeben. Die Meldung erhält den Zeitstempel des Bedien- und Beobachtungssystems.

Meldenummernverfahren

Das Meldenummernverfahren erfordert nur einen Arbeitsschritt durch den Programmierer:

- Er erstellt am PG das Anwenderprogramm, setzt das gewünschte Bit und ordnet unmittelbar beim Programmieren dem Bit den gewünschten Meldetext zu.



Eine zyklische Abfrage des Automatisierungssystems findet nicht statt. Sobald das AS eine Änderung meldet, wird die entsprechende Meldenummer an das Bedien- und Beobachtungssystem weitergereicht und der zugehörige Meldetext ausgegeben. Die Meldung hat den Zeitstempel des AS und ist daher genauer zuzuordnen als beim Bitmeldeverfahren.

14.1.3 Auswahl des Meldeverfahrens

Übersicht

Die folgende Tabelle zeigt die Eigenschaften und Randbedingungen der unterschiedlichen Meldeverfahren.

Meldenummernverfahren	**Bitmeldeverfahren**
• Meldungen werden in einer für PG und Bediensystem gemeinsamen Datenbasis verwaltet • Die Busbelastung ist gering (AS meldet aktiv) • Meldungen erhalten Zeitstempel des Automatisierungssystems	• Es gibt keine gemeinsame Datenbasis für PG und Bediensystem • Die Busbelastung ist hoch (Bediensystem pollt) • Meldungen erhalten Zeitstempel des Bediensystems

Das Meldenummernverfahren kennt die folgenden drei Arten von Meldungen:

Bausteinbezogene Meldungen	**Symbolbezogene Meldungen**	**Anwenderdefinierte Diagnosemeldungen**

• programmsynchron
• Anzeige über WinCC und ProTool (nur ALARM_S)
• Möglich bei S7-300/400
• Programmierung über Meldebausteine:
  • ALARM
  • ALARM_8
  • ALARM_8P
  • NOTIFY
  • ALARM_S(Q)
  • AR_SEND
• Übertragung in das Bediensystem
  • für WinCC über AS-OS-Verbindungsprojektierung
  • für ProTool über ProTool-Funktionen
	• programmasynchron
	• Anzeige über WinCC
	• nur möglich bei S7-400
	• Projektierung über die Symboltabelle
	• Übertragung in das AS über Systemdaten bausteine (SDBs)
	• Übertragung in das Bediensystem über AS-OS-Verbindungsprojektierung
		• programmsynchron
		• Anzeige im Diagnosepuffer auf dem PG
		• möglich bei S7-300/400
		• Programmierung über Meldebaustein (Systemfunktion)
		  • WR_USMSG
		• keine Übertragung in das Bediensystem

STEP 7 unterstützt das komfortablere Meldenummernverfahren, das im weiteren Verlauf ausführlich beschrieben wird.

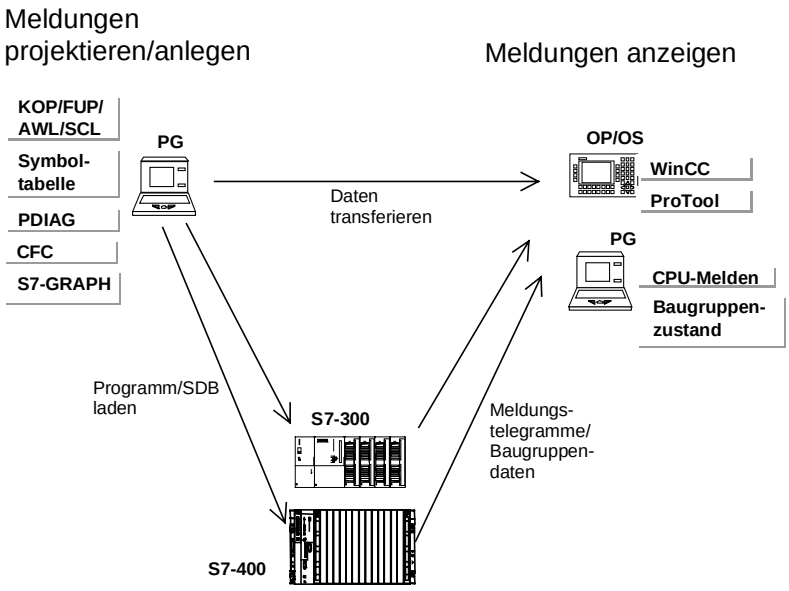
Beispiele für das Meldenummernverfahren

Meldeverfahren	**Anwendungsbereich**
Bausteinbezogene Meldungen	Zum Melden von programmsynchronen Ereignissen, z. B. soll angezeigt werden, daß ein Regler einen Grenzwert erreicht hat
Symbolbezogene Meldungen	Zum Melden von Ereignissen, die programmunabhängig sind, z. B. soll eine Schaltereinstellung überwacht werden
Anwenderdefinierte Meldungen	Zum Melden von Diagnoseereignissen im Diagnosepuffer, bei jedem Aufruf der SFC

14.1.4 SIMATIC-Komponenten

Übersicht

Das folgende Bild zeigt in einer Übersicht, welche SIMATIC-Komponenten beim Projektieren und Anzeigen von Meldungen beteiligt sind.



14.1.5 Bestandteile einer Meldung

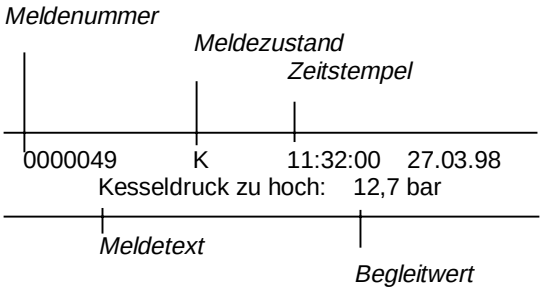
Wie eine Meldung angezeigt wird, ist abhängig vom Meldeverfahren, vom verwendeten Meldebaustein und vom Anzeigegerät.

Mögliche Bestandteile sind in der folgenden Tabelle aufgelistet:

Bestandteil	Beschreibung
Zeitstempel	Wird beim Auftreten des Meldeereignisses im Automatisierungssystem erzeugt
Meldezustand	Möglich sind: gekommen, gegangen, gegangen ohne Quittierung, gegangen mit Quittierung
Begleitwert	Einigen Meldungen kann ein Prozeßwert beigefügt werden, der vom verwendeten Meldebaustein ausgewertet werden kann
Abbild	Bei einem Systemabsturz können die angefallenen Meldungen im Nachhinein an der OS angezeigt werden.
Meldenummer	Projektweit eindeutige Nummer, die vom System vergeben wird und eine Meldung identifiziert.
Meldetext	Wird vom Anwender projektiert

Beispiel

Das folgende Beispiel zeigt eine Störmeldung an einem Operator Panel.



14.1.6 Vergabe von Meldenummern

Meldungen sind durch eine projektweit eindeutige Nummer identifiziert. Zu diesem Zweck wird den einzelnen S7-Programmen je ein Nummernbereich innerhalb des zur Verfügung stehenden Gesamtbereichs (1 - 2097151) zugeordnet. Kopiert man ein Programm, so muß im Konfliktfall - wenn im Zielbereich bereits die gleichen Meldenummern vergeben sind - dem neuen Programm ein anderer Nummernbereich zugeordnet werden. STEP 7 öffnet in einem solchen Fall automatisch das Dialogfeld, in dem Sie den neuen Nummernbereich angeben können.

Außerdem können Sie, wenn im Programm noch keine Meldungen projektiert sind, mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldenummern** den Nummernbereich für ein S7-Programm festlegen oder ändern.

Die Meldenummernbereiche werden defaultmäßig in 20.000er Schritten vergeben.

14.2 Bausteinbezogene Meldungen zuordnen und bearbeiten

14.2.1 Bausteinbezogenen Meldungen zuordnen und bearbeiten

Bausteinbezogene Meldungen werden einem Baustein (FB) zugeordnet. Für das Erstellen einer bausteinbezogenen Meldung können Sie Systemfunktionsbausteine (SFBs) und Systemfunktionen (SFCs) als Meldebausteine verwenden.

14.2.2 Welche Meldebausteine gibt es?

Sie haben die Auswahl zwischen den folgenden Meldebausteinen, in denen bereits eine Meldefunktion programmiert ist:

- SFB 33: ”ALARM”
- SFB 34: ”ALARM_8”
- SFB 35 ”ALARM_8P”
- SFB 36 ”NOTIFY”
- der SFC 18: ”ALARM_S” und SFC 17: ”ALARM_SQ” sowie
- SFB 37: ”AR_SEND” (zum Senden von Archiven).

Nähere Informationen finden Sie in der Referenzhilfe zu Bausteinen .

Wann setzen Sie welchen Meldebaustein ein?

Die folgende Tabelle hilft Ihnen bei der Entscheidung, welchen Meldebaustein Sie für Ihren Anwendungsfall auswählen. Die Auswahl des Meldebausteins ist abhängig von:

- der Anzahl der im Baustein zur Verfügung stehenden Kanäle und damit die Anzahl der Signale, die je Bausteinaufruf überwacht werden,
- der Möglichkeit, Meldungen zu quittieren
- der Möglichkeit, Begleitwerte mitzugeben.
- den Anzeigegeräten, die eingesetzt werden sollen
- dem Mengengerüst Ihrer CPU.

Melde-baustein	Ka-näle	Quittie-rung	Begleit-werte	Anzeige WinCC	Anzeige PRO-TOOL	Anzeige CPU-Melden/ S7-Status	AS	Besonderheiten
ALARM SFB 33	1	möglich	bis zu 10	ja	nein	nein	S7-400	Sendet bei kommender oder gehender Flanke je eine Meldung
ALARM_8 SFB 34	8	möglich	nein	ja	nein	nein	S7-400	Sendet bei kommender oder gehender Flanke eines oder mehrerer Signale eine Meldung
ALARM_8P SFB 35	8	möglich	bis zu 10	ja	nein	nein	S7-400	wie ALARM_8
NOTIFY SFB 36	1	nein	bis zu 10	ja	nein	nein	S7-400	wie ALARM
AR_SEND SFB 37	1	-	-	ja	nein	nein	S7-400	dient zum Senden eines Archivs
ALARM_SQ SFC 17	1	möglich	1	ja	ja*	ja	S7-300/400	Nicht ein Flanken-wechsel, sondern jeder SFC-Aufruf erzeugt eine Meldung
ALARM_S SFC 18	1	nein	1	ja	ja*	ja	S7-300/400	wie ALARM_SQ
* je nach OP-Typ								

14.2.3 Formalparameter, Systemattribute und Meldebausteine

Formalparameter als Meldenummerneingang

Für jede Meldung oder Gruppe von Meldungen benötigen Sie in Ihrem Programm einen Formalparameter, den Sie als Eingangsvariable in der Variablendeklarationstabelle Ihres Programms angeben. Der Formalparameter wird als Meldenummerneingang benutzt und bildet die Basis für eine Meldung.

Versorgung der Formalparameter mit Systemattributen

Als Voraussetzung für den Einstieg in die Meldungsprojektierung müssen Sie die Formalparameter mit Systemattributen versorgt haben:

1. Sie haben folgende Systemattribute für Parameter angefügt: "S7_server" und "S7_a_type"
2. Den Systemattributen haben Sie Werte zugewiesen, die den Meldebausteinen entsprechen, die Sie in Ihrem Programmcode aufgerufen haben. Der Wert für "S7_server" ist immer "alarm_archiv", der Wert für "S7_a_type" entspricht dem aufgerufenen Meldebaustein.

Systemattribute und zugehörige Meldebausteine

Als Objekte des Meldungsmanagers werden nicht die Meldebausteine selbst angezeigt, sondern die zugehörigen Werte der Systemattribute "S7_a_type". Diese Werte haben die gleichen Bezeichnungen wie die Meldebausteine, die als SFB oder SFC vorhanden sind (Ausnahme: "alarm_s").

S7_a_type	Meldebaustein	Bezeichnung	Eigenschaften
alarm_8	ALARM_8	SFB 34	8 Kanäle, Quittierung möglich, keine Begleitwerte
alarm_8p	ALARM_8P	SFB 35	8 Kanäle, Quittierung möglich, bis zu 10 Begleitwerte pro Kanal
notify	NOTIFY	SFB 36	1 Kanal, keine Quittierung, bis zu 10 Begleitwerte
alarm	ALARM	SFB 33	1 Kanal, Quittierung möglich, bis zu 10 Begleitwerte
alarm_s	ALARM_S	SFC 18	1 Kanal, keine Quittierung, bis zu 1 Begleitwert
alarm_s	ALARM_SQ	SFC 17	1 Kanal, Quittierung möglich, bis zu 1 Begleitwert
ar_send	AR_SEND	SFB 37	dient zum Senden eines Archivs

Nähere Informationen finden Sie in der Referenzhilfe zu Systemattributen .

Die Systemattribute werden automatisch zugewiesen, wenn die Meldebausteine, die Sie in Ihrem Programm verwenden, SFBs oder FBs mit entsprechenden Systemattributen sind und als Multiinstanzen aufgerufen werden.

14.2.4 Meldungstyp und Meldungen

Mit der Meldungsprojektierung können Sie in verschiedenen Arbeitsgängen entweder einen Meldungstyp oder Meldungen erstellen. Dies hängt von dem meldefähigen Baustein ab, über den Sie in die Meldungsprojektierung einsteigen.

Der meldefähige Baustein kann ein FB oder ein Instanz-DB sein

- Bei einem FB können Sie einen Meldungstyp erstellen, der als Vorlage für Meldungen dient. Alle Eingaben,die Sie für den Meldungstyp machen, werden automatisch in die Meldungen übernommen. Wenn Sie dem FB einen Instanz-DB zuordnen, werden nach der Vorlage des Meldungstyps automatisch Meldungen zum Instanz-DB generiert und Meldungsnummern zugeordnet.
- Bei einem Instanz-DB können Sie die Meldungen, die aus dem Meldungstyp generiert wurden, instanzspezifisch modifizieren.

Der sichtbare Unterschied ist, daß bei den Meldungen Meldungsnummern zugeordnet wurden, bei einem Meldungstyp dagegen noch nicht.

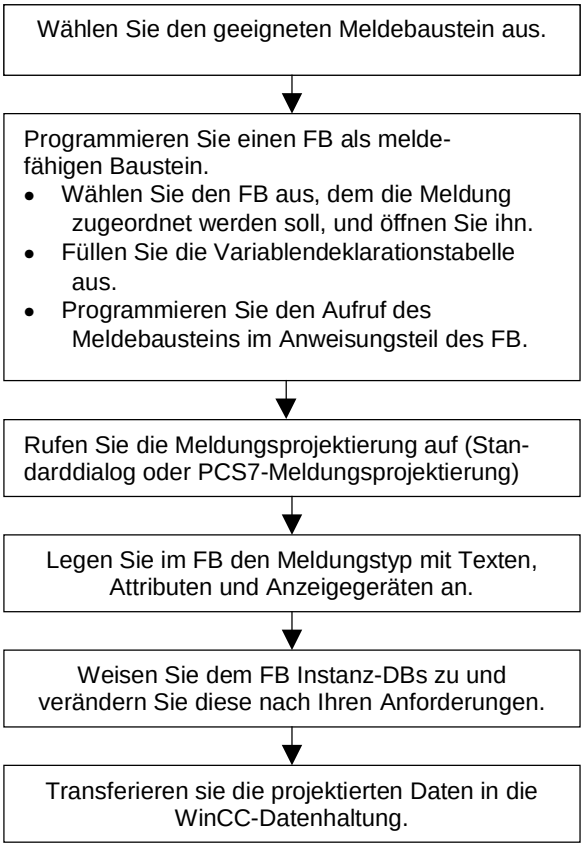
Verriegelung der Daten beim Meldungstyp

Mit der Meldungsprojektierung geben Sie Texte und Attributen zu ereignisabhängigen Meldungen ein. Dabei können Sie z. B. festlegen, wie die Meldungen auf bestimmten Anzeigegeräten angezeigt werden. Zur Erleichterung der Erstellung der Meldungen können Sie mit Vorlagen, den Meldungstypen, arbeiten.

- Wenn Sie Daten (Attribute und Texte) für den Meldungstyp eingeben, können Sie festlegen, ob sie verriegelt werden sollen oder nicht. Bei verriegelten Attributen wird ein Schlüsselsymbol neben das Eingabefeld gesetzt. Verriegelte Texte erhalten ein Häkchen in der Spalte “Gesperrt”.
- Im Meldungstyp verriegelte Daten können Sie in den instanzspezifischen Meldungen nicht mehr ändern. Die Daten werden nur angezeigt.
- Wenn Sie dennoch Änderungen benötigen, müssen Sie wieder zum Meldungstyp zurück gehen, die Verriegelung wieder aufheben, und die Änderungen dort durchführen. Die Änderungen gelten allerdings nicht für Instanzen, die vor der Änderung generiert wurden.

14.2.5 Bausteinbezogene Meldungen anlegen

Prinzipielle Vorgehensweise


Wählen Sie den geeigneten Meldebaustein aus.
Programmieren Sie einen FB als melde-fähigen Baustein.
• Wählen Sie den FB aus, dem die Meldung zugeordnet werden soll, und öffnen Sie ihn.
• Füllen Sie die Variablendeklarationstabelle aus.
• Programmieren Sie den Aufruf des Meldebausteins im Anweisungsteil des FB.
Rufen Sie die Meldungsprojektierung auf (Stan-darddialog oder PCS7-Meldungsprojektierung)
Legen Sie im FB den Meldungstyp mit Texten, Attributen und Anzeigegeräten an.
Weisen Sie dem FB Instanz-DBs zu und verändern Sie diese nach Ihren Anforderungen.
Transferieren sie die projektierten Daten in die WinCC-Datenhaltung.

Meldefähigen Baustein (FB) programmieren

1. Wählen Sie im SIMATIC Manager den Funktionsbaustein (FB) aus, für den Sie eine bausteinbezogene Meldung anlegen wollen und öffnen Sie diesen Baustein durch Doppelklick.

   Ergebnis: Der angewählte Baustein wird im Fenster ”KOP/AWL/FUP” geöffnet und angezeigt.

2. Füllen Sie die Variablendeklarationstabelle aus. Für jeden Meldebaustein, der im FB aufgerufen wird, müssen Sie im aufrufenden FB Variablen deklarieren.

   Tragen Sie dazu in der Variablendeklarationstabelle in der Spalte ”Deklaration” folgende Variablen ein:

   - beim Deklarationstyp ”in” einen symbolischen Namen für den Meldebausteineingang, z. B. ”Meld01” (für Meldeeingang 01) und den Typ (muß ”DWORD” ohne Anfangswert sein).
   - beim Deklarationstyp ”stat” einen symbolischen Namen für den aufzurufenden Meldebaustein, z. B. ”alarm” und den entsprechenden Typ, hier also ”SFB33”.

3. Fügen Sie im Anweisungsteil des FBs den Aufruf für den ausgewählten Meldebaustein ein, hier also ”CALL alarm”, und schließen Sie Ihre Eingabe ab mit der Taste RETURN.

   Ergebnis: Im Anweisungsteil des FBs werden die Eingangsvariablen des aufgerufenen Meldebausteins, hier also des SFB 33, angezeigt.

4. Weisen Sie der Variable ”EV_ID” den von Ihnen in Schritt 2 vergebenen symbolischen Namen für den Meldebausteineingang, hier also ”Meld01”, zu, und bestätigen Sie, daß die Systemattribute für die Meldungsprojektierung übernommen werden sollen.

   Ergebnis: In der Spalte ”Name” erscheint, falls diese nicht selektiert ist, ein ”Fähnchen”. Damit ist der ausgewählte Baustein meldefähig. Die benötigten Systemattribute (z. B. S7_server und S7_a_type) und die entsprechenden Werte werden automatisch zugewiesen.

5. Wiederholen Sie die Schritte 2. bis 4. für alle Aufrufe von Meldebausteinen in diesem FB.
6. Speichern Sie den Baustein mit dem Menübefehl **Datei > Speichern**.
7. Schließen Sie das Fenster ”KOP/AWL/FUP”.

Meldungsprojektierung aufrufen

- Wählen Sie im SIMATIC Manager den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

  Ergebnis: Das Dialogfeld für die STEP 7-Meldungsprojektierung (Standarddialog) wird geöffnet. Was Sie tun müssen, um die PCS7-Meldungs-projektierung aufzurufen, lesen Sie unter PCS7-Meldungsprojektierung.

Meldungstyp anlegen

1. Selektieren Sie den gewünschten Meldebaustein und tragen Sie in den Registern ”Attribute” und ”Text” die gewünschten Meldungsattribute und den gewünschten Meldungstext ein.
   Falls Sie einen mehrkanaligen Meldebaustein (z. B. ”ALARM_8”) selektiert haben, können Sie jeder Subnummer einen eigenen Meldungstext zuweisen. Die Attribute gelten für alle Subnummern.
2. Weisen Sie dem Meldungstyp die gewünschten Anzeigegeräte zu, indem Sie auf die Schaltfläche ”Neues Gerät” klicken und im nachfolgenden Dialogfeld ”Anzeigegerät einfügen”die gewünschten Anzeigegeräte auswählen.

Tragen Sie dazu in den folgenden Registerblättern die gewünschten Texte und Attribute für die Anzeigegeräte ein. Verlassen Sie das Dialogfeld mit ”OK”.

Hinweis

Zum Bearbeiten der anzeigegerätespezifischen Texte und Attribute beachten Sie bitte die mit Ihrem Anzeigegerät mitgelieferte Dokumentation.

Instanz-DBs anlegen

1. Nachdem Sie einen Meldungstyp angelegt haben, können Sie diesem nun Instanz-Datenbausteine (DBs) zuweisen und die Meldungen für diese DBs instanzspezifisch bearbeiten.
   Öffnen Sie dazu im SIMATIC Manager den Baustein, der Ihren zuvor projektierten FB aufrufen soll, z. B. ”OB1”, indem Sie darauf doppelklicken. Tragen Sie im geöffneten Anweisungsteil des OBs den Aufruf (”CALL”), sowie den Namen und die Nummer des aufzurufenden FBs und des DBs ein, den Sie dem FB als Instanz zuweisen wollen. Schließen Sie Ihre Eingabe ab mit der Taste RETURN.

   Beispiel: Tragen Sie ”CALL FB1, DB1” ein. Falls der DB1 noch nicht existiert, so bestätigen Sie die Frage, ob der Instanz-DB generiert werden soll, mit ”Ja”.

   Ergebnis: Der Instanz-DB wird angelegt. Im Anweisungsteil des OBs werden die Eingangsvariablen des zugehörigen FBs, hier z. B. ”Meld01” und die vom System vergebene Meldungsnummer, hier die ”1”, angezeigt.
2. Speichern Sie den OB mit dem Menübefehl **Datei > Speichern** und schließen Sie das Fenster ”KOP/AWL/FUP”.

Meldungen bearbeiten

1. Selektieren Sie im SIMATIC Manager den erzeugten Instanz-DB, z. B. ”DB1” und rufen Sie mit dem Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung** die Meldungsprojektierung auf.

   Ergebnis: Das Dialogfeld ”Meldungsprojektierung” wird geöffnet und der selektierte Instanz-DB mit der vom System zugewiesenen Meldungsnummer wird angezeigt.

2. Tragen Sie die gewünschten Änderungen für den jeweiligen Instanz-DB in den verschiedenen Registerblättern ein und fügen Sie weitere Anzeigegeräte hinzu, falls gewünscht. Verlassen Sie die Funktion mit ”OK”.

   Ergebnis: Damit ist die Meldungsprojektierung für den angewählten Instanz-DB abgeschlossen.

Projektierungsdaten transferieren

- Transferieren Sie die projektierten Daten in die Datenhaltung von WinCC (über die AS-OS-Verbindungsprojektierung) bzw. in die Datenhaltung von ProTool.

14.2.6 PCS7-Meldungsprojektierung

Für das Bearbeiten von Meldungstypen und Meldungen, die auf WinCC-Anzeigegeräten ausgegeben werden sollen, bietet STEP 7 mit der PCS7-Meldungsprojektierung eine komfortable Möglichkeit

- die Projektierung von Anzeigegeräten zu vereinfachen (werden automatisch angelegt)
- die Eingabe von Attributen und Texten für Meldungen zu vereinfachen
- die Einheitlichkeit von Meldungen zu gewährleisten.

Aufrufen der PCS7-Meldungsprojektierung

1. Selektieren Sie im SIMATIC Manager den Baustein (FB oder DB), dessen Meldetexte Sie bearbeiten wollen und rufen Sie mit dem Menübefehl **Bearbeiten > Objekteigenschaften** das Dialogfeld für die Eingabe der Systemattribute auf.
2. Tragen Sie in die angezeigte Tabelle eines der folgenden Systemattribut ein:
   - Attribut: ”S7_alarm_ui” und Wert: ”1” für nicht ereignisgesteuerte PCS 7-Meldungen
     (der Wert "0" schaltet die PCS 7-Meldungsprojektierung aus).
     Eigenschaften-Parameter können in KOP/FUP/AWL vergeben werden. Anschließend generierte und dem entsprechenden FB zugeordnete DBs übernehmen die Einstellungen und können über die eigenen Attributeinstellungen unabhängig vom Meldungstyp (FB) umgeschaltet werden.

- Attribut: "S7_alarm" und Wert ""alarm_16" (wenn der selektierte Baustein ein ereignisgesteuerter Kommunikationsbaustein vom Typ EDC ist) Eigenschaften-Parameter dürfen bei ereignisgesteuerter Kommunikation nicht vergeben werden. Die Meldungstexte der diesem FB zugeordneten DBs können im SIMATIC Manager nicht editiert werden.

Hinweis

Beim Eintragen der Systemattribute wird eine Syntaxprüfung durchgeführt und die fehlerhaften Eingaben werden rot gekennzeichnet.

3. Verlassen Sie das Dialogfeld mit ”OK”.
4. Wählen Sie den Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Meldung**.

   Ergebnis: Das Dialogfeld für die PCS7-Meldungsprojektierung wird geöffnet.

Bearbeiten von Meldungstypen

1. Selektieren Sie im SIMATIC Manager den FB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS7-Meldungsprojektierung auf.
   Ergebnis: Im Dialogfeld wird für jeden Meldebaustein, für den Sie im FB eine Variable deklariert haben, ein Registerblatt bzw. für den ereignisgesteuerten Kommunikationsbaustein zwei Registerblätter angezeigt.
2. Füllen Sie die Textfelder für die Meldungsbestandteile "Herkunft", "OS-Bereich" und "Batch-Kennung" aus.
3. Geben Sie für alle Ereignisse der verwendeten Meldebausteine die Meldeklasse und den Ereignistext ein und legen Sie fest, ob jedes Ereignis einzeln quittiert werden muß.
4. Klicken Sie für die Meldungsbestandteile, die für alle Instanzen gelten und dort nicht verändert werden sollen, das Kontrollkästchen "Gesperrt" an.

Bearbeiten von Meldungen

1. Selektieren Sie im SIMATIC Manager den Instanz- DB, dessen Meldetexte Sie bearbeiten wollen und rufen Sie die PCS7-Meldungsprojektierung auf.
2. Verändern Sie die nicht gesperrten instanzspezifischen Meldungsbestandteile.

Hinweis

Die Meldetexte von Instanz-DBs, die ereignisgesteuerten Kommunikationsbausteinen zugeordnet sind, können nur in CFC bearbeitet werden.

14.3 Symbolbezogene Meldungen zuordnen und bearbeiten

14.3.1 Symbolbezogene Meldungen zuordnen und bearbeiten

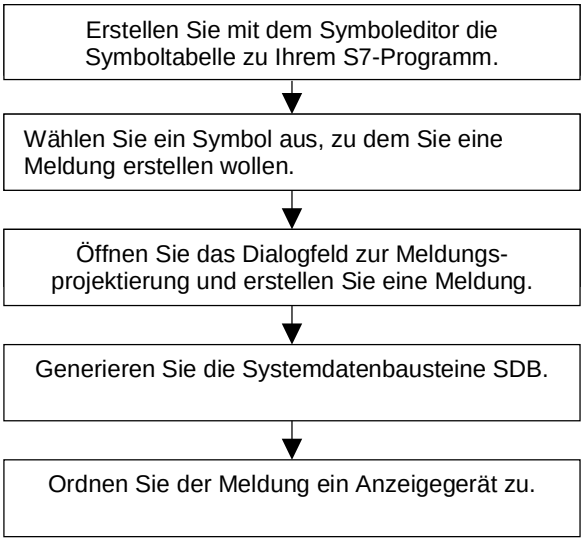
Symbolbezogenen Meldungen (SCAN) werden direkt einem Signal in der Symboltabelle zugeordnet. Zulässige Signale sind alle booleschen Operanden: also Eingänge (E), Ausgänge (A) und Merker (M). Diesen Signalen können Sie in der Meldungsprojektierung verschiedene Attribute, Meldungstexte und bis zu 10 Begleitwerte zuordnen. Die Auswahl von Signalen aus der Symboltabelle wird Ihnen dadurch erleichtert, daß Sie Filter einstellen können.

Mit einer symbolbezogenen Meldung können Sie in einem vorgegebenen Zeitraster ein Signal abtasten, um zu erfassen, ob ein Signalwechsel vorliegt.

Hinweis

Das Zeitraster ist abhängig von der verwendeten CPU.

Prinzipielle Vorgehensweise


Erstellen Sie mit dem Symboleditor die Symboltabelle zu Ihrem S7-Programm.
Wählen Sie ein Symbol aus, zu dem Sie eine Meldung erstellen wollen.
Öffnen Sie das Dialogfeld zur Meldungs-projektierung und erstellen Sie eine Meldung.
Generieren Sie die Systemdatenbausteine SDB.
Ordnen Sie der Meldung ein Anzeigegerät zu.

Während des Ablaufs werden asynchron zu Ihrem Programm die Signale überprüft, für die Sie Meldungen projektiert haben. Die Überprüfungen finden in den projektierten Zeitrastern statt. Die Meldungen werden auf den zugeordneten Anzeigegeräten angezeigt.

14.4 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten

14.4.1 Anwenderdefinierte Diagnosemeldungen anlegen und bearbeiten

Mit dieser Funktion können Sie einen Anwendereintrag in den Diagnosepuffer schreiben und eine entsprechende Meldung senden, die Sie in der Meldungsprojektierung erstellen. Anwenderdefinierte Diagnosemeldungen werden durch die System-Funktion SFC 52 (WR_USMSG) realisiert, die als Meldebaustein genutzt wird. Den Aufruf für die SFC 52 müssen Sie in Ihr Anwenderprogramm einfügen und ihn mit der Ereignis-ID versorgen.

Anwenderdefinierte Diagnosemeldungen können im Gegensatz zu den baustein- und den symbolbezogenen Meldungen nur auf einem PG angezeigt werden. Deshalb können Sie diesen Meldungen in der Meldungsprojektierung keine Anzeigegeräte zuordnen.

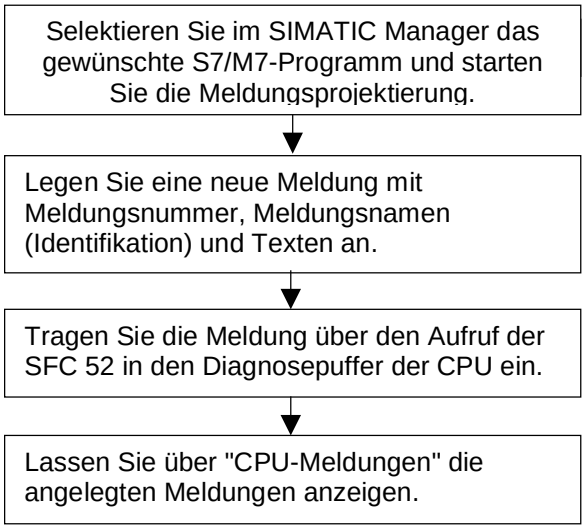
Voraussetzungen

Bevor Sie eine anwenderdefinierte Diagnosemeldung anlegen können, müssen Sie

- im SIMATIC Manager ein Projekt angelegt haben und
- im Projekt das S7/M7-Programm erzeugt haben, dem Sie die Meldung zuordnen wollen.

Prinzipielle Vorgehensweise

Um eine anwenderdefinierte Diagnosemeldung anzulegen und anzeigen zu lassen, gehen Sie so vor:


Selektieren Sie im SIMATIC Manager das gewünschte S7/M7-Programm und starten Sie die Meldungsprojektierung.
Legen Sie eine neue Meldung mit Meldungsnummer, Meldungsnamen (Identifikation) und Texten an.
Tragen Sie die Meldung über den Aufruf der SFC 52 in den Diagnosepuffer der CPU ein.
Lassen Sie über "CPU-Meldungen" die angelegten Meldungen anzeigen.

14.5 Bedienerrelevante Texte übersetzen und bearbeiten

14.5.1 Bedienerrelevante Texte übersetzen und bearbeiten

Texte, die während der Prozeßbearbeitung auf Anzeigegeräten ausgegeben werden, werden üblicherweise in der Sprache eingegeben, in der die Automatisierungslösung programmiert wurde.

Häufig tritt jedoch der Fall ein, daß ein Bediener, der später auf die Meldungen reagieren soll, diese Sprache nicht beherrscht. Er benötigt die Texte in seiner Muttersprache. Nur so können einwandfreie Bearbeitung und schnelle Reaktion auf die ausgegebenen Meldungen gewährleistet werden.

STEP 7 bietet Ihnen die Möglichkeit, sämtliche bedienerrelevanten Texte in jede beliebige Sprache zu übersetzen. Einzige Voraussetzung ist, daß Sie die Sprache in Ihrem Projekt bereits installiert haben (Menübefehl im SIMATIC Manager: **Extras > Sprache für Anzeigegeräte**). Die Anzahl der angebotenen Sprachen wird bei der Installation von Windows festgelegt (Systemeigenschaft).

Sie können so sicherstellen, daß jeder, der zu einem späteren Zeitpunkt mit den ausgegebenen Meldungen konfrontiert wird, diese Texte in der von ihm gewünschten Sprache angezeigt bekommt. Auf diese Weise erhöhen Sie die Sicherheit im Prozeß ganz wesentlich.

Bedienerrelevante Texte sind Anwendertexte und Textbibliotheken.

14.5.2 Anwendertexte übersetzen und bearbeiten

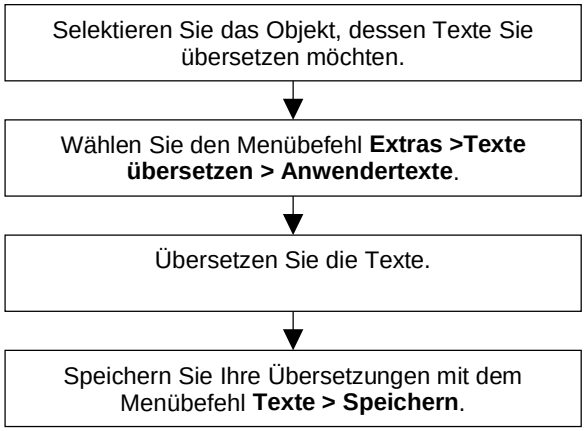
Anwendertexte können für ein ganzes Projekt, für S7-Programme, den Baustein-Ordner oder einzelne Bausteine und für die Symboltabelle angelegt werden, sofern Meldungen in diesen Objekten projektiert sind. Sie enthalten alle Texte und Meldungen, die z. B. auf Anzeigegeräten angezeigt werden können. Zu einem Projekt kann es mehrere Listen bedienerrelevanter Texte geben, die Sie in die erforderlichen Sprachen übersetzen können.

Die in einem Projekt verfügbaren Sprachen können Sie auswählen (Menübefehl: **Extras > Sprache für Anzeigegeräte**). Sie können auch nachträglich Sprachen hinzufügen oder löschen.

Wenn Sie die Anwendertexte eines STEP 7-Objektes öffnen (Menübefehl: **Extras > Texte übersetzen > Anwendertexte**), wird auf dem Bildschirm eine Tabelle angezeigt, deren Spalten jeweils eine Sprache darstellen. In der ersten Spalte wird immer die als Standardsprache eingestellte Sprache angezeigt.

Prinzipielle Vorgehensweise

Stellen Sie sicher, daß Sie im SIMATIC Manager mit dem Menübefehl **Extras > Sprache für Anzeigegeräte** die Sprachen eingestellt haben, in die Sie die Anwendertexte übersetzen wollen.


Selektieren Sie das Objekt, dessen Texte Sie übersetzen möchten.
Wählen Sie den Menübefehl **Extras >Texte übersetzen > Anwendertexte**.
Übersetzen Sie die Texte.
Speichern Sie Ihre Übersetzungen mit dem Menübefehl **Texte > Speichern**.

Exportieren und Importieren bedienerrelevanter Texte

Sie können in STEP 7 erzeugte bedienerrelevante Texte außerhalb von STEP 7 übersetzen oder bearbeiten. Dazu exportieren Sie die angezeigte Liste bedienerrelevanter Texte in Textdateien im CSV-Format, die Sie mit einem ASCII-Editor oder einem Tabellenbearbeitungswerkzeug, z. B. der Microsoft-Applikation EXCEL, bearbeiten können. Anschließend importieren Sie die Texte wieder in STEP 7.

Bedienerrelevante Texte können nur in den Projektteil importiert werden, aus dem sie exportiert wurden.

14.5.3 Textbibliotheken übersetzen

Textbibliotheken stellen eine Liste von Texten zur Verfügung, die in Meldungen integriert, dynamisch zur Laufzeit aktualisiert und am PG oder anderen Anzeigegeräten angezeigt werden können. Sie sind CPUs zugeordnet. Die Texte in System-Textbibliotheken werden von STEP 7 oder STEP 7-Optionspaketen bereitgestellt. Zu einer CPU kann es mehrere Textbibliotheken geben, die Sie in die erforderlichen Sprachen übersetzen können.

Die in einem Projekt verfügbaren Sprachen können Sie im SIMATIC Manager auswählen (Menübefehl: **Extras > Sprache für Anzeigegeräte**). Sie können auch nachträglich Sprachen hinzufügen oder löschen.

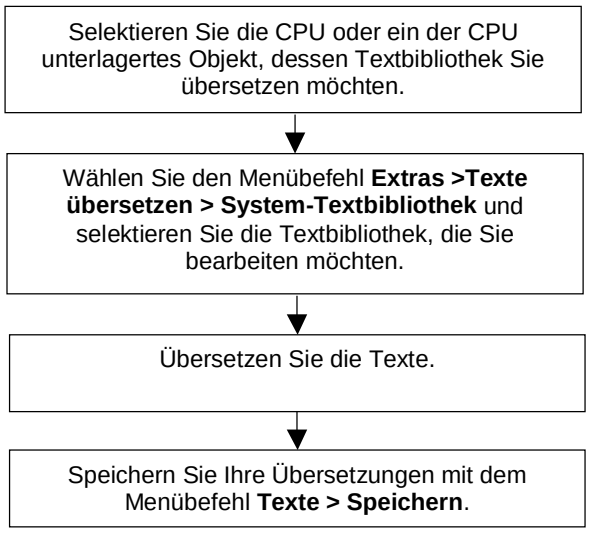
Wenn Sie eine Textbibliothek öffnen (Menübefehl **Extras > Texte übersetzen > System-Textbibliothek**), wird auf dem Bildschirm eine Tabelle angezeigt, deren Spalten jeweils eine Sprache darstellen. In der ersten Spalte wird immer der Index angezeigt, mit dem die einzelnen Texte referenziert werden können.

Beispiel

Index	deutsch	französisch
1732	ausgefallen	ne répond pas
1733	gestört	défaillant
1734	Parametrierfehler	erreur de paramétrage

Prinzipielle Vorgehensweise

Stellen Sie sicher, daß Sie im SIMATIC Manager mit dem Menübefehl **Extras > Sprache für Anzeigegeräte** die Sprachen eingestellt haben, in die Sie eine Textbibliothek übersetzen wollen.


Selektieren Sie die CPU oder ein der CPU unterlagertes Objekt, dessen Textbibliothek Sie übersetzen möchten.
Wählen Sie den Menübefehl **Extras >Texte übersetzen > System-Textbibliothek** und selektieren Sie die Textbibliothek, die Sie bearbeiten möchten.
Übersetzen Sie die Texte.
Speichern Sie Ihre Übersetzungen mit dem Menübefehl **Texte > Speichern**.

14.6 Meldungsprojektierungsdaten zum Zielsystem transferieren

14.6.1 Projektierungsdaten zum Zielsystem transferieren

Einleitung

Mit dem Transferprogramm AS-OS-Engineering transferieren Sie die erstellten Meldungsprojektierungsdaten in den Datenbestand von WinCC.

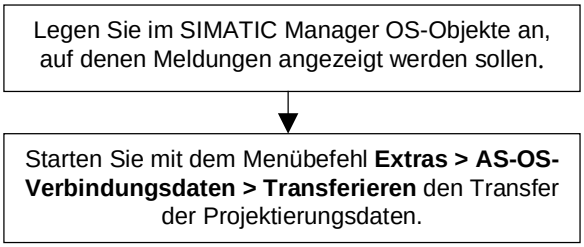
Sie haben dabei die Möglichkeit, verschiedene Transferoptionen auszuwählen. So können Sie z. B. einen Adreß- und Textabgleich durchführen, um sicherzustellen, daß aktuelle Daten übertragen werden.

Voraussetzungen

Bevor Sie mit dem Transfer beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben das Setup-Programm AS-OS-Verbindungsprojektierung ausgeführt.
- Sie haben die Projektierungsdaten für das Anlegen von Meldungen erzeugt.

Prinzipielle Vorgehensweise


Legen Sie im SIMATIC Manager OS-Objekte an, auf denen Meldungen angezeigt werden sollen.
Starten Sie mit dem Menübefehl **Extras > AS-OS-Verbindungsdaten > Transferieren** den Transfer der Projektierungsdaten.

14.7 CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen

14.7.1 CPU-Meldungen und Anwenderdefinierte Diagnosemeldungen anzeigen

Mit der Funktion ”CPU-Meldungen” können asynchrone Meldungen von Diagoseereignissen und anwenderdefinierte Diagnosemeldungen und oder ALARM_S/SQ-Meldungen ausgegeben werden.

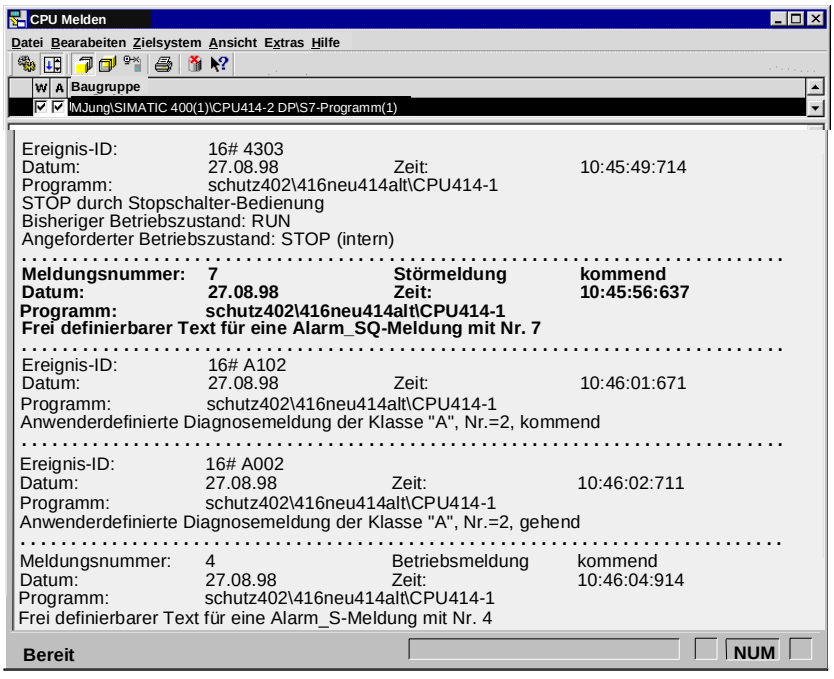
Außerdem können Sie von CPU-Melden aus mit dem Menübefehl **Extras > Meldungen projektieren** die Meldungsprojektierung starten und anwenderdefinierte Diagnosemeldungen anlegen. Voraussetzung dafür ist, daß Sie CPU-Melden über ein Online-Projekt gestartet haben.

Anzeigemöglichkeiten

Mit der Funktion ”CPU-Meldungen” können Sie entscheiden, ob und wie Meldungen für ausgewählte CPUs angezeigt werden:

- **"In den Vordergrund holen"**: Das Fenster mit den CPU-Meldungen erscheint im Vordergrund und wird dort bei jeder neuen Meldung aufgeblendet.
- **"Im Hintergrund lassen"**: Das Empfangen der CPU-Meldungen läuft im Hintergrund. Das Fenster bleibt bei neuen Meldungen im Hintergrund und kann bei Bedarf in den Vordergrund geholt werden.
- **"Meldung ignorieren"**: CPU-Meldungen werden **nicht** angezeigt und im Gegensatz zu den ersten beiden Fällen **nicht** archiviert.

Im Fenster ”CPU Melden” können Sie die im Archiv enthaltenen Meldungen durchblättern. Einige Beispiele sind im folgenden Bild dargestellt:



Quittierbare Meldungen (ALARM_SQ) werden in fetter Schrift angezeigt und können mit dem Menübefehl **Bearbeiten > CPU-Meldung quittieren** quittiert werden.

Archivfunktion

Zur Archivierung der Meldungen steht Ihnen ein Archiv zur Verfügung, in dem zwischen 40 und 2000 CPU-Meldungen archiviert werden können. Ist die eingestellte Archivgröße überschritten, wird jeweils die älteste vorhandene Meldung gelöscht.

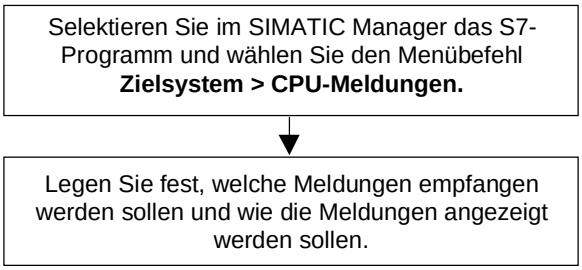
Aktualisieren von ALARM_S/SQ-Meldungen

Beim Aktualisieren von ALARM_S/SQ-Meldungen werden alle nicht gegangenen und/oder nicht quittierten Meldungen erneut in das Archiv eingetragen. Aktualisiert werden die Meldungen

- beim Wiederanlauf der Baugruppe, auf die sich die Meldungen beziehen (nicht bei Neustart)
- beim Konfigurieren der CPU-Meldungen, wenn Sie im Dialogfeld "Einstellungen - CPU Melden" das Feld "A" für ALARM_S/SQ anklicken.

Prinzipielle Vorgehensweise

Um CPU-Meldungen für ausgewählte Baugruppen zu konfigurieren,


Selektieren Sie im SIMATIC Manager das S7-Programm und wählen Sie den Menübefehl **Zielsystem > CPU-Meldungen.**
Legen Sie fest, welche Meldungen empfangen werden sollen und wie die Meldungen angezeigt werden sollen.

14.7.2 CPU-Meldungen konfigurieren

Um CPU-Meldungen für ausgewählte Baugruppen zu konfigurieren, gehen Sie folgendermaßen vor:

1. Starten Sie im SIMATIC Manager die Applikation CPU-Melden über ein Online-Projekt. Markieren Sie dazu online ein S7-Programm und rufen Sie mit dem Menübefehl **Zielsystem > CPU Meldungen** die Applikation CPU-Melden für die ausgewählte CPU auf.

   Ergebnis: Es erscheint das Applikationsfenster ”CPU-Melden”, das die angemeldete CPU auflistet.

2. Melden Sie bei Bedarf weitere CPUs an, indem Sie Schritt 1 für weitere Programme oder Schnittstellen wiederholen.

3. Klicken Sie die Kontrollkästchen vor den Listeneinträgen an und legen Sie fest, welche Meldungen für die Baugruppe empfangen werden sollen:

   A: aktiviert ALARM_SQ (SFC17) und ALARM_S (SFC 18) z. B Prozessdiagnosemeldungen von S7-PDIAG; S7-GRAPH oder Systemfehler melden.

   W: aktiviert Diagnoseereignisse.

4. Legen Sie die Archivgröße fest.

   Ergebnis: Sobald die o.g. Meldungen auftreten, werden Sie in der von Ihnen festgelegten Form in das Meldungsarchiv geschrieben und angezeigt.

Hinweis

In der Liste der angemeldeten Baugruppen im Applikationsfenster "CPU-Melden" sind die CPUs eingetragen, für die Sie den Menübefehl **Zielsystem > CPU-Meldungen** aufgerufen haben. Die Einträge in der Liste bleiben erhalten, bis sie im oben genannten Dialogfeld gelöscht werden.

14.7.3 Anzeigen von gespeicherten CPU-Meldungen

CPU-Meldungen werden immer im Archiv aufgezeichnet, sofern Sie nicht den Menübefehl **Ansicht > Meldung ignorieren** gewählt haben. Es werden immer alle archivierten Meldungen angezeigt.

14.8 'Melden von Systemfehlern' projektieren

14.8.1 'Melden von Systemfehlern' projektieren

Einführung

S7-Komponenten und DP-Normslaves können bei Eintritt eines Systemfehlers Aufrufe von Organisationsbausteinen auslösen.

Beispiel: bei Drahtbruch kann eine diagnosefähige Baugruppe Diagnosealarm auslösen (OB 82).

Die S7-Komponenten stellen Informationen zum aufgetretenen Systemfehler zur Verfügung. Die Startereignisinformationen, d.h. die Lokaldaten des zugeordneten OBs (enthalten u.a. den Datensatz 0), geben allgemeine Auskunft über Ort (z.B. logische Adresse der Baugruppe) und Art (z.B. Kanalfehler oder Pufferungsausfall) des Fehlers.

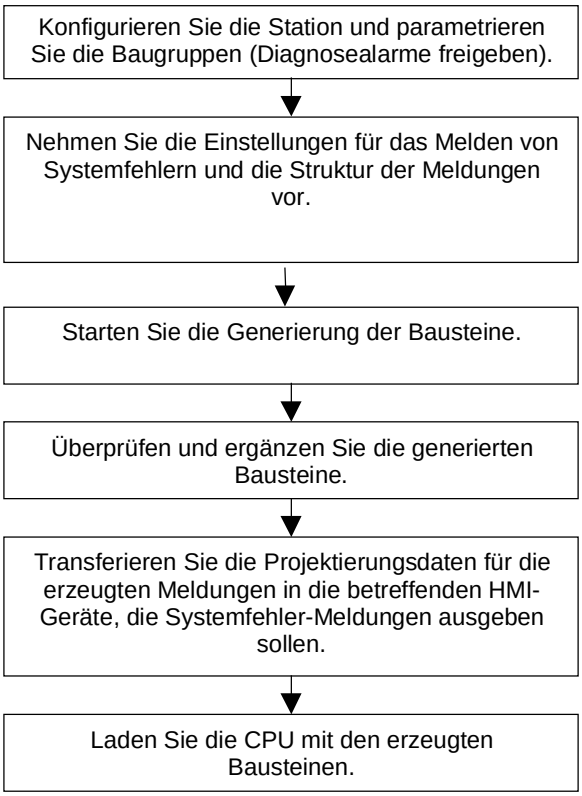
Darüber hinaus kann über zusätzliche Diagnoseinformationen (Lesen von Datensatz 1 mit SFC51 bzw. Lesen des Diagnosetelegramms von DP-Normslaves mit SFC13) der Fehler genauer spezifiziert werden: z.B. Kanal 0 oder 1, Drahtbruch oder Meßbereichsüberschreitung.

STEP 7 bietet mit der Funktion "Systemfehler melden" eine komfortable Möglichkeit, die von der Komponente zur Verfügung gestellten Diagnoseinformationen in Form von Meldungen anzuzeigen.

Die hierfür notwendigen Bausteine und Meldetexte werden von STEP 7 automatisch erzeugt. Vom Anwender müssen die erzeugten Bausteine lediglich in die CPU geladen und die Texte in angeschlossene HMI-Geräte transferiert werden.

Eine genaue Übersicht über die unterstützten Diagnose-Informationen bei verschiedenen Slaves finden Sie unter Abschnitt Unterstützte Komponenten und Funktionsumfang.

Prinzipielle Vorgehensweise


Konfigurieren Sie die Station und parametrieren Sie die Baugruppen (Diagnosealarme freigeben).
Nehmen Sie die Einstellungen für das Melden von Systemfehlern und die Struktur der Meldungen vor.
Starten Sie die Generierung der Bausteine.
Überprüfen und ergänzen Sie die generierten Bausteine.
Transferieren Sie die Projektierungsdaten für die erzeugten Meldungen in die betreffenden HMI-Geräte, die Systemfehler-Meldungen ausgeben sollen.
Laden Sie die CPU mit den erzeugten Bausteinen.

Die Meldungen werden über den Standardmeldeweg ALARM_S/SQ an "CPU-Meldungen" am PG oder an die angeschlossenen HMI-Geräte gesendet.

14.8.2 Unterstützte Komponenten und Funktionsumfang

Die Komponenten der S7-300-Stationen, S7-400-Stationen und DP-Slaves werden von "Systemfehler melden" unterstützt, sofern sie Funktionen wie Diagnosealarm, Ziehen/Stecken-Alarm und kanalspezifische Diagnose unterstützen.

Folgende Komponenten werden von "Systemfehler Melden" **nicht** unterstützt:

- M7-, C7 und PROFIBUS-DP-Konfigurationen über DP-Masteranschaltung (CP 342-5 DP) in S7-300 Stationen
- H-Systeme
- SIMATIC PC-Stationen (Soft-PLC, Slot-PLC)

Im Falle eines Wiederanlaufes ist noch zusätzlich zu beachten, daß es zu fehlenden Alarmmeldungen kommen kann. Dieses Verhalten beruht darauf, daß der Meldequittierspeicher der CPU bei einem Wiederanlauf nicht gelöscht wird, aber "Systemfehler melden" die internen Daten zurücksetzt.

In den beiden unteren Tabellen finden Sie alle Diagnoseblöcke von den verschiedenen Slaves, die von "Systemfehler melden" unterstützt werden:

Diagnose-block	**Kennungsbe-zeichnung** (fehlerhafter Steckplatz)	**Kanalbe-zeichnung** (Kanalfehler) 1)	**Modulstatus** (Modulfehler, falsches/kein Modul)	**Hersteller-spezifisch**	**Gerätebe-zeichnung** (nur bei ET 200 B)
Header-kennung 2)	**0x01**	**0x10**	**0x00 Typ 0x82**	**0x00 Typ > 0x9F**	**0x00 + 1 Byte Diagnoseinfo**
ET 200 S	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	Klartextmeldung aus GSD-Datei 3)	- 4)
ET 200 M als S7 Slave	wird nicht ausgewertet	wird nicht ausgewertet	wird nicht ausgewertet	-	-
ET 200 M als Norm Slave	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-	-
ET 200 X	Meldung: "Diagnose liegt vor"	-	-	-	-
ET 200 X DESINA	Meldung: "Diagnose liegt vor"	Klartextmeldung	Klartextmeldung	-	-
ET 200 L	Meldung: "Diagnose liegt vor"	-		-	-
ET 200 B Digital	Meldung: "Diagnose liegt vor"	-	-	-	Meldung: "Baugruppe gestört"
ET 200 B Analog	Meldung: "Diagnose liegt vor"	-	-	-	

1) Standardmeldungen nach DPV0, bei Norm Slaves zusätzlich aus GSD-Datei

2) Headerkennung: Kennung im Diagnosetelegramm, die unterschiedliche Diagnosteile identifiziert.

3) GSD-Datei: Beschreibung eines DPV0 bzw. DPV1 Norm Slaves in Dateiform

4) "-" bedeutet, daß die Komponente die Information nicht liefert.

Diagnoseblock	**DS0/DS1** 1)	**H-Status.** (H-Systeme werden noch nicht unterstützt)	**sonst. Ausprägung**
Headerkennung 2)	**0x00 Typ 0x01**	**0x00 Typ = 0x9E oder Typ = 0x9F**	**0x00 sonst. Typ**
ET 200 S	Klartextmeldung	- 3)	wird nicht ausgewertet
ET 200 M als S7 Slave	Klartextmeldung	wird nicht ausgewertet	wird nicht ausgewertet
ET 200 M als Norm Slave	Klartextmeldung	wird noch nicht ausgewertet	wird nicht ausgewertet
ET 200 X	Meldung: "Baugruppe gestört"	-	wird nicht ausgewertet
ET 200 X DESINA	Klartextmeldung	-	wird nicht ausgewertet
ET 200 L	Meldung: "Baugruppe gestört"	-	-
ET 200 B Digital	-	-	-
ET 200 B Analog	Klartextmeldung	-	wird nicht ausgewertet

1) DS0: Standard-Diagnose, z. B. Baugruppenstörung, externe Hilfsspannung oder Frontstecker fehlt, Umfang 4 Byte, in den Lokaldaten des OB 82 enthalten.
DS1: Kanalfehler, für jeden Kanaltyp unterschiedlich definiert, lesbar im Anwenderprogramm über SFC 51. Die Texte kommen aus der S7-HW-Diagnose.

2) Headerkennung: Kennung im Diagnosetelegramm, die unterschiedliche Diagnosteile identifiziert.

3) "-" bedeutet, daß die Komponente die Information nicht liefert.

Das Diagnosetelegramm (auch Norm Slave Telegramm genannt) ist aus den oben genannten Diagnoseblöcken aufgebaut und kann über SFC 13 im Anwenderprogramm gelesen werden.

In STEP 7 wird das Diagnosetelegramm über den Aufruf des Baugruppenzustands im Online-Fenster ”HWKonfig” (Hardware diagnostizieren) im Register "DP-Slave Diagnose" unter "Hexdarstellung" angezeigt.

14.8.3 Einstellungen für das Melden von Systemfehlern

Sie haben mehrere Möglichkeiten, den Dialog für die Einstellungen aufzurufen:

- Selektieren Sie in HW Konfig die CPU, für welche Sie dase Melden von Systemfehlern projektieren möchten. Wählen Sie dann den Menübefehl **Extras > Systemfehler melden**.
- Wenn Sie bereits Bausteine für das Melden von Systemfehlern erzeugt haben, dann können Sie den Dialog durch Doppelklicken auf einen erzeugten Baustein (FB, DB) aufrufen.
- Wählen Sie die Option zum automatischen Aufruf bei Speichern und Übersetzen der Konfiguration im Eigenschaftsdialog der Station

Die Option zum automatischen Aufruf bei Speichern und Übersetzen erreichen Sie wie folgt:

1. Markieren Sie im SIMATIC Manager die betreffende Station.
2. Wählen Sie den Menübefehl **Bearbeiten > Objekteigenschaften**.
3. Wählen Sie das Register "Einstellungen".

Im Dialog geben Sie unter anderem ein:

- Welcher FB und welcher zugeordnete Instanz-DB erzeugt werden soll.
- das CPU-Verhalten im Fehlerfall: Sie können einstellen, ob die CPU nach dem Melden eines Systemfehlers in STOP übergeht oder nicht.
- Erzeugen von Fehler-OBs: ob im S7-Programm noch nicht vorhandenen Fehler-Obs erzeugt werden sollen oder nicht.
- ob Meldungen quittierbar sein sollen.
- ob Referenzdaten erzeugt werden sollen.
- ob Warnungen während der Generierung von Systemfehler melden stets angezeigt werden sollen
- ob der Dialog erscheinen soll, wenn Systemfehler melden automatisch nach Speichern und Übersetzen der Konfiguation aufgerufen wird (siehe Einstellung oben).
- das Erscheinungsbild der Meldungen (Aufbau und Reihenfolge der möglichen Textanteile).

Detaillierte Informationen finden Sie in der Hilfe zum aufgerufenen Dialog.

14.8.4 Bausteine für die Meldung von Systemfehlern generieren

Um die benötigten Bausteine (FB und DB, je nach Einstellung auch noch nicht vorhandenen OBs) zu erzeugen, gehen Sie folgendermaßen vor:

- Klicken Sie im Dialogfeld "Systemfehler melden" die Schaltfläche "Generieren" an.

Folgende Bausteine werden erzeugt:

- Fehler-OBs (wenn das Kontrollkästchen "Fehler-OBs erzeugen" angeklickt wurde)
- Diagnose-FB (voreingestellt: FB 49)
- Instanz-DB für den Diagnose-FB (voreingestellt: DB 49)
- Optionaler Anwender-FB, der von dem Diagnose-FB aufgerufen wird

14.8.5 Erzeugte Fehler-OBs

Haben Sie im Register "Allgemein" des Dialogfeldes "Systemfehler melden" das Kontrollkästchen "Fehler-OBs erzeugen" angeklickt, werden beim Generieren folgende Fehler-OBs erzeugt:

- OB 81 (Stromversorgungsfehler) mit Aufruf des erzeugten Diagnose-FBs.
- OB 82 (Diagnosealarm-OB; nur wenn Baugruppen oder DP-Slaves projektiert wurden) mit Aufruf des erzeugtenDiagnose-FBs.
- OB 83 (Ziehen/Stecken-Alarm) mit Aufruf des erzeugten Diagnose-FBs.
- OB 84 (CPU-Hardwarefehler)
  Dieser OB wird ohne Inhalt erzeugt, damit die CPU bei Kommunikationsfehlern (z. B. bei Problemen mit MPI-Abschlußwiderstand bei Ziehen und Stecken des MPI-Kabels) nicht in STOP geht. Fehler werden nicht ausgewertet, es wird keine Meldung erzeugt.
- OB 85 (Programmablauffehler)
  Nur bei Fehlern bei der Prozeßabbildaktualisierung (z. B. Ziehen der Baugruppe) wird der STOP der CPU verhindert - damit Diagnose-FB in OB 83 bearbeitet werden kann. Eine evtl. Einstellung „CPU-STOP nach Meldung von "Systemfehler melden“ wird im OB 83 wirksam.
  Bei allen anderen Fehlerereignissen von OB 85geht die CPU in STOP.
- OB 86 (Ausfall eines Erweiterungsbaugruppenträgers, eines DP-Mastersystems oder eines Dezentralen Peripheriegerätes) mit Aufruf des erzeugten Diagnose-FBs.
  Dieser OB wird nur erzeugt, wenn eine der genannten Komponenten projektiert wurde.

Wenn die Fehler-OBs bereits existieren...

Vorhandene Fehler-OBs werden nicht überschrieben. Sie müssen dann den Aufruf des erzeugten FBs im entsprechenden OB selbst einfügen.

Wenn die Konfiguration Dezentrale Peripheriegeräte umfaßt...

Für die Auswertung von Fehlern in der Dezentralen Peripherie ruft der erzeugte FB automatisch die SFC 13 auf (Diagnosedaten der DP-Slaves lesen). Um diese Funktion sicherzustellen, muß der erzeugte FB entweder nur im OB 1 oder in einem Weckalarm-OB mit kurzem Zeittakt und in den Anlauf-OBs aufgerufen werden.

ACHTUNG:

Bitte beachten Sie, daß:

- durch den von "Systemfehler melden" erzeugten OB 85 bei dem Fehlerereignis "Fehler beim Aktualisieren des Prozeßabbilds" KEIN STOP der CPU mehr erfolgt;
- der OB 85 von der CPU zusätzlich auch in den Fehlerfällen
  - "Fehlerereignis für einen nicht geladenen OB"
  - "Fehler beim Aufruf bzw. Zugriff auf einen nicht geladenen Baustein"

  aufgerufen wird. In diesen Fehlerfällen erfolgt mit dem von "Systemfehler melden" erzeugten OB 85 weiterhin ein STOP der CPU wie vor Verwendung von "Systemfehler melden";
- die Einstellung "CPU geht nach Ausführung des Diagnose-FB in STOP“ ist für OB 84 und OB 85 NICHT wirksam, weil in diesen OBs der FB von "Systemfehler melden" nicht aufgerufen wird. Im Falle des OB 85 wird diese Einstellung indirekt vom FB-Aufruf im OB 83 beachtet.

14.8.6 Erzeugter FB, DB

Der erzeugte FB wertet die Lokaldaten des Fehler-OBs aus, liest evtl. zusätzliche Diagnoseinformationen der Fehler auslösenden S7-Komponente, und erzeugt automatisch die entsprechende Meldung.

Er hat folgende Eigenschaften:

- Erstellungssprache SFM (Systemfehler Melden) (gilt ebenfalls für den erzeugten Instanz-DB)
- Know-How-protected (gilt ebenfalls für den erzeugten Instanz-DB)
- Verzögert während der Laufzeit ankommende Alarme
- Ruft durch Doppelklick den Dialog zum Einstellen der Funktion "Melden von Systemfehlern" auf (gilt ebenfalls für den erzeugten Instanz-DB).

Anwender-Baustein

Weil der Diagnose-FB Know-how-geschützt ist, können Sie ihn nicht editieren. Der FB stellt aber eine Schnittstelle für das Anwenderprogramm zur Verfügung, so daß Sie Zugriff auf z. B. den Fehlerstatus oder die Meldungsnummer haben.

Der Baustein für die Auswertung im Anwenderprogramm (im Register "Anwender-Baustein" des Dialogs einstellbar) wird im erzeugten FB mit den folgenden Parametern aufgerufen:

```
EV_C : BOOL ;          //Meldung kommend (TRUE) oder gehend (FALSE)
EV_ID : DWORD ;        //generierte Meldungsnummer
IO_Flag: BYTE ;        //Eingabebaugruppe: B#16#54 Ausgabebaugruppe:
                       //B#16#55
logAdr : WORD ;        //logische Adresse
TextlistId : WORD ;    //ID der Textliste (Standardliste = 1)
ErrorNo : WORD ;       //generierte Fehlernummer
Channel_Error : BOOL ; //Kanalfehler (TRUE)
ChannelNo : WORD ;     //Kanalnummer
```

Falls der Anwender-FB noch nicht existiert, wird er von SFM mit den oben aufgelisteten Parametern angelegt.

15 Bedienen und Beobachten von Variablen

15.1 Projektieren von bedien- und beobachtbaren Variablen

Übersicht

STEP 7 bietet Ihnen eine komfortable Methode, um variable Größen Ihres Prozesses bzw. Ihres Automatisierungssystems mit WinCC bedien- und beobachtbar zu machen.

Der Vorteil gegenüber bisherigen Methoden liegt für Sie darin, daß Sie das Projektieren der Daten nicht mehr für jede Operator Station (OS) einzeln, sondern nur einmalig mit STEP 7 durchführen müssen. Die beim Projektieren mit STEP 7 erzeugten Daten können Sie mit Hilfe des Transferprogramms AS-OS-Engineering (Bestandteil des SW-Paketes ”Process Control System PCS7”) in die Datenhaltung von WinCC übertragen, wobei die Konsistenz und Verträglichkeit der Daten mit dem Anzeigesystem überprüft wird. WinCC verwendet die Daten in Bildbausteinen und Grafikobjekten.

Mit STEP 7 können Sie BuB-Attribute für folgende Variablen projektieren bzw. ändern:

- Ein-, Ausgangs- und Durchgangsparameter von Funktionsbausteinen.
- Merker und E/A-Signale
- Parameter von CFC-Bausteinen in CFC-Plänen.

Prinzipielle Vorgehensweise

Die Vorgehensweise bei der Projektierung von bedien- und beobachtbaren Variablen ist abhängig von der gewählten Programmier- bzw. Projektiersprache und der Art der Variablen, die Sie bedienen und beobachten möchten. Die prinzipielle Vorgehensweise umfaßt aber stets die folgenden Schritte:

1. Weisen Sie den Parametern eines Funktionsbausteins oder den Symbolen in einer Symboltabelle Systemattribute für das Bedienen und Beobachten zu.

   Bei CFC entfällt dieser Schritt, da Sie die Bausteine schon vorbereitet aus einer Bibliothek entnehmen.

2. Versehen Sie die Variablen, die Sie bedienen und beobachten wollen, in einem Editierdialog mit den benötigten BuB-Attributen (S7_m_c). Im Dialogfeld "Bedienen und Beobachten" (Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Bedienen und Beobachten**) können Sie Win-CC-Attribute wie Grenzwerte, Ersatzwerte und Protokollierungseigenschaften, usw verändern.

3. Transferieren Sie die mit STEP 7 erzeugten Projektierungsdaten mit Hilfe des AS-OS-Engineering in Ihr Anzeigesystem (WinCC).

Namenskonventionen

Damit die Projektierungsdaten für WinCC gespeichert und transferiert werden können, werden sie unter einem vom STEP 7 automatisch vergebenen, eindeutigen Namen abgelegt. Die Namen der bedien- und beobachtbaren Variablen, der CFC-Pläne und der S7-Programme bilden Bestandteile dieses Namens und unterliegen aus diesem Grund bestimmten Konventionen:

- Die Namen der S7-Programme in einem S7-Projekt müssen eindeutig sein (unterschiedliche Stationen dürfen keine S7-Programme mit gleichem Namen enthalten).
- Die Namen der Variablen, S7-Programme und CFC-Pläne dürfen keine Unterstriche, Leerzeichen oder die folgenden Sonderzeichen: [‘] [.] [%] [-] [/] [*] [+] enthalten.

15.2 Projektieren von BuB-Attributen mit AWL, KOP und FUP

15.2.1 Projektieren von BuB-Attributen mit AWL, KOP und FUP

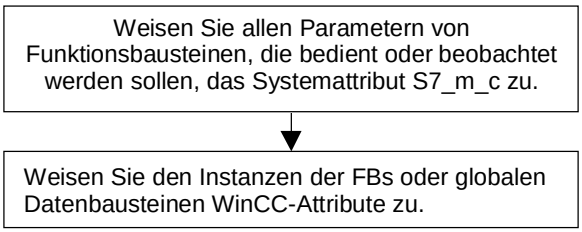
Einleitung

Mit der hier beschriebenen Vorgehensweise können Sie Parameter von Funktionsbausteinen bedien- und beobachtbar machen und zugeordneten Instanz-DBs oder Global-Datenbausteinen in Ihrem Anwenderprogramm die erforderlichen BuB-Attribute zuweisen.

Voraussetzung

Sie haben ein STEP 7-Projekt, ein S7-Programm und einen FB angelegt.

Prinzipielle Vorgehensweise


Weisen Sie allen Parametern von Funktionsbausteinen, die bedient oder beobachtet werden sollen, das Systemattribut S7_m_c zu.
Weisen Sie den Instanzen der FBs oder globalen Datenbausteinen WinCC-Attribute zu.

15.3 Projektieren von BuB-Attributen über die Symboltabelle

15.3.1 Projektieren von BuB-Attributen über die Symboltabelle

Einleitung

Unabhängig von der eingesetzten Programmiersprache projektieren Sie mit der hier beschriebenen Vorgehensweise folgende Variablen:

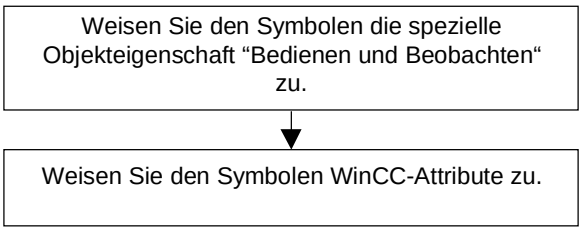
- Merker
- E/A-Signale

Voraussetzung

Bevor Sie beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben im SIMATIC Manager ein Projekt angelegt.
- In diesem Projekt muß ein S7-Programm mit einer Symboltabelle vorhanden sein.
- Die Symboltabelle muß geöffnet sein.

Prinzipielle Vorgehensweise


Weisen Sie den Symbolen die spezielle Objekteigenschaft “Bedienen und Beobachten“ zu.
Weisen Sie den Symbolen WinCC-Attribute zu.

15.4 Abändern von BuB-Attributen mit CFC

15.4.1 Abändern von BuB-Attributen mit CFC

Einleitung

Mit CFC erstellen Sie Ihr Anwenderprogramm, indem Sie Bausteine, die bereits bedien- und beobachtbar sind, aus einer Bibliothek auswählen und in einem Plan plazieren und verschalten.

Voraussetzung

Sie haben in einem STEP 7-Projekt ein S7-Programm eingefügt, einen CFC-Plan angelegt und darin Bausteine plaziert.

Prinzipielle Vorgehensweise

Bearbeiten Sie die Objekteigenschaften der Bausteine.

Hinweis

Verwenden Sie selbsterstellte Bausteine, denen Sie in der Erstellsprache das Systemattribut S7_m_c zugewiesen haben, können Sie diese Bausteine bedien- und beobachtbar machen, indem Sie im Dialogfeld "Bedienen und Beobachten" (Menübefehl **Bearbeiten > Spezielle Objekteigenschaften > Bedienen und Beobachten**) das Optionsfeld "Bedienen und Beobachten" aktivieren.

15.5 Transferieren der Projektierungsdaten zum BuB-Zielsystem

15.5.1 Transferieren der Projektierungsdaten zum BuB-Zielsystem

Einleitung

Mit dem Transferprogramm AS-OS-Engineering transferieren Sie die erstellten Projektierungsdaten für das Bedienen und Beobachten in den Datenbestand von WinCC.

Sie haben dabei die Möglichkeit, verschiedene Transferoptionen auszuwählen. So können Sie z. B. einen Adreß- und Textabgleich durchführen, um sicherzustellen, daß aktuelle WinCC-Attribute übertragen werden.

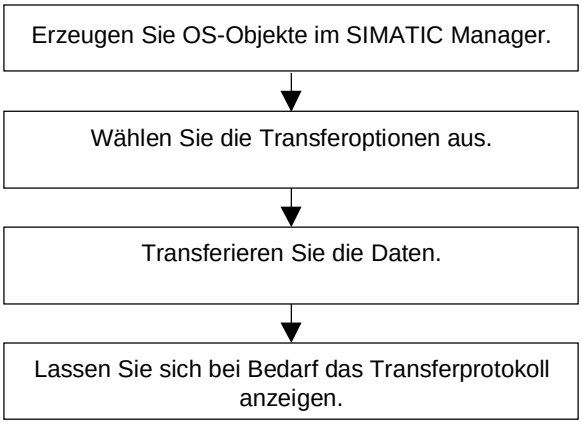
Voraussetzung

Bevor Sie mit dem Transfer beginnen, müssen folgende Voraussetzungen erfüllt sein:

- Sie haben das Setup-Programm AS-OS-Verbindungsprojektierung (Engineering) ausgeführt.
- Sie haben die Projektierungsdaten für das Bedienen und Beobachten erzeugt.

Prinzipielle Vorgehensweise

Um die Projektierungsdaten für das Bedienen und Beobachten in den Datenbestand von WinCC zu transferieren, gehen Sie folgendermaßen vor:


Erzeugen Sie OS-Objekte im SIMATIC Manager.
Wählen Sie die Transferoptionen aus.
Transferieren Sie die Daten.
Lassen Sie sich bei Bedarf das Transferprotokoll anzeigen.

16 Aufbau der Online-Verbindung und CPU-Einstellungen

16.1 Aufbau von Online-Verbindungen

16.1.1 Aufbau von Online-Verbindungen

Eine Online-Verbindung zwischen Erstellsystem und Zielsystem ist für das Laden von S7-Anwenderprogrammen/Bausteinen, das Zurückladen von Bausteinen aus dem S7-Zielsystem in das Erstellsystem und für folgende weitere Tätigkeiten erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU
- Anzeigen und Einstellen von Uhrzeit und Datum der CPU
- Anzeigen des Baugruppenzustands
- Baustein on-/offline vergleichen
- Hardware diagnostizieren

Um eine Online-Verbindung herstellen zu können, müssen Erstellsystem und Zielsystem über eine geeignete Schnittstelle miteinander verbunden sein (z.B. mehrpunktfähige Schnittstelle (MPI)). Sie können dann über das Online-Fenster des Projekts oder über das Fenster "Erreichbare Teilnehmer" auf das Zielsystem zugreifen.

16.1.2 Online-Verbindung aufbauen über das Fenster "Erreichbare Teilnehmer"

Diese Art bietet die Möglichkeit des schnellen Zugriffs, z.B. für Servicezwecke. Sie können auf alle im Netz erreichbaren programmierbaren Baugruppen zugreifen. Wählen Sie diese Methode, wenn keine Projektdaten zu den Zielsystemen auf ihrem PG vorhanden sind.

Das Fenster "Erreichbare Teilnehmer" läßt sich mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** aufrufen. Unter dem Objekt "Erreichbare Teilnehmer" werden alle im Netz erreichbaren programmierbaren Baugruppen mit ihrer Adresse angezeigt.

Es können auch Teilnehmer angezeigt werden, die nicht mit STEP 7 programmierbar sind (z.B. Programmiergeräte oder Operator-Panels).

16.1.3 Online-Verbindung aufbauen über das Online-Fenster des Projekts

Wählen Sie diese Methode, wenn Sie das Zielsystem in einem Projekt auf Ihrem PG/PC projektiert haben. Das Online-Fenster läßt sich im SIMATIC Manager mit dem Menübefehl **Ansicht > Online** aufrufen. Es veranschaulicht die Projektdaten auf dem Zielsystem (im Gegensatz zum Offline-Fenster, das die Projektdaten auf dem PG/PC veranschaulicht). Im Online-Fenster werden sowohl für das S7-Programm als auch für das M7-Programm die Daten angezeigt, die auf dem Zielsystem liegen.

Diese Sicht auf das Projekt benutzen Sie für Zugriffe auf das Zielsystem. So sind gewisse Funktionen im Menü "Zielsystem" des SIMATIC Managers zwar im Online-Fenster, nicht aber im Offline-Fenster aufrufbar.

Man unterscheidet:

- **Zugriff mit konfigurierter Hardware**
  Damit können Sie nur auf die offline projektierten Baugruppen zugreifen. Auf welche der online erreichbaren Baugruppen zugegriffen wird, liegt durch die MPI-Adresse fest, die beim Konfigurieren der programmierbaren Baugruppe eingestellt wurde.
- **Zugriff ohne konfigurierte Hardware**
  Dafür ist ein vorhandenes S7-Programm bzw. M7-Programm Voraussetzung, das unabhängig von der Hardware erstellt wurde (d.h. direkt unter dem Projekt liegt). Auf welche der online erreichbaren Baugruppen zugegriffen wird, wird hier durch Angabe der entsprechenden MPI-Adresse in den Objekteigenschaften des S7-/M7-Programms festgelegt.

Der Zugriff über das Online-Fenster kombiniert die Daten auf dem Zielsystem mit den zugehörigen Daten auf dem Erstellsystem. Wenn Sie zum Beispiel einen S7-Baustein unter einem Projekt online öffnen, setzt sich die Anzeige wie folgt zusammen:

- Codeteil des Bausteins aus der CPU des S7-Zielsystems und
- Kommentaren und der Symbolik aus der Datenhaltung des Erstellsystems (sofern offline vorhanden). Wenn Sie Bausteine ohne vorhandene Projektstruktur direkt in der angeschlossenen CPU öffnen, werden diese angezeigt, wie sie in der CPU sind, d.h. ohne Symbolik und Kommentare.

16.1.4 Paßwortschutz für Zugriff auf Zielsysteme

Mit Hilfe des Paßwortschutzes können Sie

- das Anwenderprogramm in der CPU mit seinen Daten vor ungewollten Änderungen schützen (Schreibschutz)
- das Know-How, das in Ihrem Anwenderprogramm enthalten ist, bewahren (Leseschutz)
- Online-Funktionen, die den Prozeß stören würden, unterbinden

Sie können eine Baugruppe nur dann mit einem Paßwort schützen, wenn die Baugruppe diese Funktionalität unterstützt.

Wenn Sie eine Baugruppe mit einem Paßwort schützen möchten, müssen Sie die Schutzstufe und das Paßwort im Rahmen der Baugruppenparametrierung definieren und anschließend die geänderte Parametrierung auf die Baugruppe laden.

Wenn für die Ausführung einer Online-Funktion die Eingabe eines Paßworts erforderlich ist, wird das Dialogfeld "Paßwort eingeben" angezeigt. Durch die Eingabe des korrekten Paßworts erhalten Sie die Zugangsberechtigung zu Baugruppen, für die im Rahmen der Parametrierung eine besondere Schutzstufe festgelegt wurde. Sie können dann Online-Verbindungen zur geschützten Baugruppe aufbauen und die zur Schutzstufe gehörigen Online-Funktionen ausführen.

Mit dem Menübefehl **Zielsystem > Zugangsberechtigung > Einrichten** können Sie das Dialogfeld zur Eingabe des Paßworts direkt aufrufen. Damit können Sie - z. B. zu Beginn einer Sitzung - das Paßwort einmalig eingeben und werden dann bei späteren Online-Zugriffen nicht mehr abgefragt. Die Paßworteingabe wirkt entweder bis zum Beenden des SIMATIC Managers oder bis zum Aufheben der Paßworteingabe mit dem Menübefehl **Zielsystem > Zugangsberechtigung > Aufheben**.

CPU-Parameter	Bemerkungen
Testbetrieb/ Prozeßbetrieb (nicht für S7-400 bzw. CPU 318-2)	Einstellbar Im Register "Schutz". Im Prozeßbetrieb werden Testfunktionen wie Programmstatus oder Status/Steuern Variable so eingeschränkt, daß die eingestellte zulässige Zykluszeiterhöhung nicht überschritten wird. So sind z. B. beim Programmstatus keine Aufrufbedingungen erlaubt und die Statusanzeige einer programmierten Schleife wird an der Rücksprungstelle abgebrochen. Das Testen mit Haltepunkten und schrittweise Programmausführung können im Prozeßbetrieb nicht ausgeführt werden. Im Testbetrieb sind alle Testfunktionen über PG/PC ohne Einschränkungen nutzbar, die auch größere Verlängerungen der Zykluszeit bewirken können.
Schutzstufe	Einstellbar im Register "Schutz". Schreibende bzw. lesende und schreibende Zugriffe auf die CPU können von der Kenntnis eines Paßwortes abhängig gemacht werden. Das Paßwort wird in diesem Register parametriert.

16.1.5 Hinweis zur Aktualisierung des Fensterinhalts

Beachten Sie:

- Änderungen im Online-Fenster eines Projekts aufgrund von Benutzeraktionen (z.B. Laden oder Löschen von Bausteinen) werden nicht automatisch in einem eventuell geöffneten Fenster "Erreichbare Teilnehmer" nachgeführt.
- Entsprechende Änderungen im Fenster "Erreichbare Teilnehmer" werden nicht automatisch in einem eventuell geöffneten Online-Fenster eines Projekts nachgeführt.

Um eine aktuelle Anzeige in einem parallel geöffneten Fenster zu erhalten, muß dieses Fenster ebenfalls aktualisiert werden (über Menübefehl oder Funktionstaste F5).

16.2 Anzeigen und Ändern des Betriebszustands

16.2.1 Anzeigen und Ändern des Betriebszustands

Mit dieser Funktion können Sie beispielsweise die CPU nach Korrektur eines Fehlers wieder in den Betriebszustand "RUN" setzen.

Anzeigen des Betriebszustands

1. Öffnen Sie Ihr Projekt und markieren Sie ein S7-/M7-Programm oder öffnen Sie mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** das Fenster "Erreichbare Teilnehmer" und markieren Sie einen Teilnehmer ("MPI=...").
2. Wählen Sie den Menübefehl **Zielsystem > Betriebszustand**.

Dieses Dialogfeld zeigt den aktuellen und den letzten Betriebszustand sowie die aktuelle Stellung des Schlüsselschalters an der Baugruppe an. Für Baugruppen, bei denen die aktuelle Schlüsselschalterstellung nicht ausgelesen werden kann, wird der Text "Undefiniert" ausgegeben.

Ändern des Betriebszustands

Über die Schaltflächen können Sie den Betriebszustand der CPU-Baugruppe ändern. Es sind nur die Schaltflächen aktiv, die im aktuellen Betriebszustand angewählt werden können.

16.3 Anzeigen und Einstellen von Uhrzeit und Datum

16.3.1 Anzeigen und Einstellen von Uhrzeit und Datum

Gehen Sie wie folgt vor:

1. Öffnen Sie Ihr Projekt und markieren Sie ein S7-/M7-Programm oder öffnen Sie mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** das Fenster "Erreichbare Teilnehmer" und markieren Sie einen Teilnehmer ("MPI=...").
2. Wählen Sie den Menübefehl **Zielsystem > Uhrzeit stellen**.
   Der Menübefehl ist wählbar, wenn im Projektfenster (Ansicht online) ein S7-/M7-Programm oder im Fenster "Erreichbare Teilnehmer" ein Teilnehmer ("MPI=...") markiert ist.
3. Im angezeigten Dialogfeld können Sie die aktuelle Uhrzeit und das aktuelle Datum auf der angewählten Baugruppe ablesen.
4. In den Eingabefeldern "Datum" und "Uhrzeit" können Sie bei Bedarf neue Werte vorgeben oder über die voreingestellte Option Uhrzeit und Datum Ihres PG/PC übernehmen.

Hinweis

Bei einer Baugruppe ohne integrierte Echtzeituhr werden als Datum 00.00.00 und als Uhrzeit 00:00:00 ausgegeben.

17 Laden

17.1 Laden aus dem PG/PC in das Zielsystem

17.1.1 Voraussetzungen für das Laden

Voraussetzungen zum Laden in das Zielsystem

- Es besteht eine Verbindung zwischen Ihrem PG und der CPU des Zielsystems (z.B. über die MPI-Schnittstelle).
- Der Zugriff auf das Zielsystem ist möglich.
- Das zu ladende Programm ist fehlerfrei übersetzt.
- Die CPU muß in einem Betriebszustand sein, bei dem das Laden zulässig ist (STOP oder RUN-P).
  Beachten Sie jedoch beim Laden im RUN-P-Zustand, daß das Programm bausteinweise übertragen wird. Überschreiben Sie dabei ein altes CPU-Programm, können Konflikte auftreten, z.B. wenn sich Bausteinparameter geändert haben. Die CPU geht dann während der Bearbeitung des Zyklus in den Betriebszustand STOP . Es wird daher empfohlen, die CPU vor dem Laden in den Betriebszustand "STOP" zu schalten.
- Falls Sie den Baustein offline geöffnet haben und laden wollen, muß der CPU-Baugruppe ein Online-Anwenderprogramm im SIMATIC Manager zugeordnet sein.
- Vor dem Laden des Anwenderprogramms, sollten Sie die CPU urlöschen, um sicherzustellen, daß sich keine “alten” Bausteine auf der CPU befinden.

Betriebszustand STOP

Setzen Sie den Betriebszustand von RUN auf STOP, bevor Sie

- das gesamte oder Teile des Anwenderprogramm auf die CPU laden,
- ein Urlöschen der CPU veranlassen,
- den Anwenderspeicher komprimieren.

Neustart (Übergang in den Betriebszustand RUN)

Wenn Sie vom Zustand "STOP" aus einen Neustart veranlassen, wird das Programm neu gestartet und zunächst im Betriebszustand "ANLAUF" das Anlauf-Programm (in Baustein OB100) abgearbeitet. Ist der Anlauf erfolgreich, wechselt die CPU in den Betriebszustand RUN. Ein Neustart ist erforderlich nach:

- Urlöschen der CPU,
- Laden des Anwenderprogramms im STOP-Zustand.

17.1.2 Unterschied zwischen Speichern und Laden von Bausteinen

Grundsätzlich ist zwischen Speichern und Laden von Bausteinen zu unterscheiden:

	Speichern	**Laden**
Menübefehle	**Datei > Speichern** Datei > Speichern unter	**Zielsystem > Laden**
Funktion	Der aktuelle Stand des Bausteins aus dem Editor wird auf der Festplatte des PGs gespeichert.	Der aktuelle Stand des Bausteins aus dem Editor wird nur in die CPU geladen.
Syntaxprüfung	Eine Syntaxprüfung wird ausgeführt. Gegebenenfalls werden Ihnen Fehler in Dialogfeldern gemeldet. Die Fehlerursache und die Fehlerstellen werden Ihnen angezeigt. Diese Fehler müssen vor dem Speichern oder Laden des Bausteins behoben sein. Bei fehlerfreier Syntax wird der Baustein anschließend in Maschinencode übersetzt und gespeichert bzw. geladen.	Eine Syntaxprüfung wird ausgeführt. Gegebenenfalls werden Ihnen Fehler in Dialogfeldern gemeldet. Die Fehlerursache und die Fehlerstellen werden Ihnen angezeigt. Diese Fehler müssen vor dem Speichern oder Laden des Bausteins behoben sein. Bei fehlerfreier Syntax wird der Baustein anschließend in Maschinencode übersetzt und gespeichert bzw. geladen

Die Tabelle gilt unabhängig davon, ob Sie den Baustein online oder offline geöffnet haben.

Hinweis zu Bausteinänderungen - erst speichern dann laden

Um neu erstellte Bausteine bzw. Änderungen im Anweisungsteil von Codebausteinen, in Deklarationstabellen oder bei Datenwerten in Datenbausteinen zu übernehmen, müssen Sie den entsprechenden Baustein abspeichern. Änderungen, die Sie im Editor durchführen und mit dem Menübefehl **Zielsystem > Laden** in die CPU transferieren, z.B. zum Testen kleiner Änderungen, sollten auf jeden Fall vor dem Beenden des Editors auch auf der Festplatte des PGs gespeichert werden. Andernfalls haben Sie anschließend unterschiedliche Stände Ihres Anwenderprogramms in der CPU und im PG. Allgemein ist zu empfehlen, immer zuerst Änderungen zu speichern und anschließend zu laden.

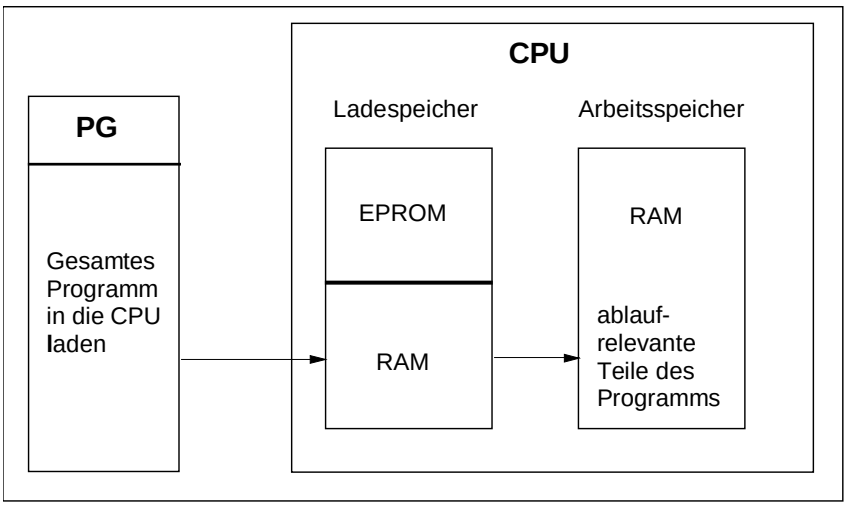
17.1.3 Lade- und Arbeitsspeicher in der CPU

Nach Abschluß der Konfiguration, Parametrierung und Programmerstellung sowie Aufbau der Online-Verbindung können Sie komplette Anwenderprogramme oder einzelne Bausteine auf ein Zielsystem übertragen. Für den Test einzelner Bausteine müssen Sie zumindest einen OB sowie die darin aufgerufenen FBs und FCs und die benutzten DBs laden. Um die nach dem Konfigurieren der Hardware, dem Projektieren von Netzen oder dem Erstellen einer Verbindungstabelle entstehenden Systemdaten auf das Zielsystem zu übertragen, laden Sie das Objekt ”Systemdatenbausteine”.

Anwenderprogramme laden Sie mit Hilfe des SIMATIC Managers in ein Zielsystem, z. B. in der Endphase des Programmtests oder um das fertige Anwenderprogramm zum Ablauf zu bringen.

Zusammenspiel von Lade- und Arbeitsspeicher der CPU

Das gesamte Anwenderprogramm wird in den Ladespeicher geladen, die ablaufrelevanten Teile des Programms auch in den Arbeitsspeicher.



Ladespeicher der CPU

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne Symboltabelle und Kommentare (diese bleiben im Speicherbereich des PG).
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Beim Ladespeicher kann es sich zielsystemabhängig um RAM-, ROM- oder EPROM-Speicher handeln.
- Bei S7-300 kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EEPROM-Anteil haben (z. B. CPU312 IFM und CPU314 IFM).
- Bei S7-400 ist der Einsatz einer Memory Card (RAM oder EEPROM) zur Erweiterung des Ladespeichers unerläßlich.

Arbeitsspeicher der CPU

Der Arbeitsspeicher (integriertes RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des Anwenderprogramms.

Mögliche Vorgehensweisen beim Laden

Über die Ladefunktion können Sie das Anwenderprogramm oder ladbare Objekte (z.B. Bausteine) in das Zielsystem laden. Ist ein Baustein schon im RAM der CPU vorhanden, so werden Sie beim Laden gefragt, ob der Baustein überschrieben werden soll oder nicht.

- Die ladbaren Objekte können Sie im Projektfenster markieren und vom SIMATIC Manager aus laden (Menübefehl: **Zielsystem > Laden**).
- Beim Programmieren von Bausteinen und beim Konfigurieren von Hardware und Netzen können Sie das gerade bearbeitete Objekt direkt über das Menü in dem zu Ihrer Tätigkeit gehörigen Hauptfenster laden (Menübefehl: **Zielsystem > Laden**).
- Eine weitere Möglichkeit ist, ein Online-Fenster mit der Sicht auf das Zielsystem zu öffnen (z.B. über **Ansicht > Online** oder **Zielsystem > Erreichbare Teilnehmer anzeigen**) und das zu ladende Objekt in das Online-Fenster zu kopieren.

Umgekehrt können Sie sich über die Ladefunktion aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

17.1.4 Lademöglichkeiten abhängig vom Ladespeicher

Aus der Aufteilung des Ladespeichers in einen RAM- und einen EEPROM-Bereich ergeben sich Konsequenzen für die Möglichkeiten beim Laden Ihres Anwenderprogramms bzw. von Bausteinen Ihres Anwenderprogramms. Um die Daten in die CPU zu laden, bestehen folgende Möglichkeiten:

Ladespeicher	Lademöglichkeiten	Kommunikationsart zwischen Erstell- und Zielsystem
RAM	Laden und Löschen von einzelnen Bausteinen	Online-Verbindung PG -Zielsystem
	Laden und Löschen eines gesamten Anwenderprogrammes	Online-Verbindung PG -Zielsystem
	Nachladen von einzelnen Bausteinen	Online-Verbindung PG -Zielsystem
EPROM integriert (nur bei S7-300) oder steckbar	Laden von gesamten Anwenderprogrammen	Online-Verbindung PG -Zielsystem
EPROM steckbar	Laden von gesamten Anwenderprogrammen	Externes Laden des EPROMS und Stecken der Memory Card oder über Online-Verbindung auf das im Zielsystem gesteckte EPROM.

Laden in das RAM über Online-Verbindung

Im Zielsystem haben Sie bei Spannungsausfall keine Datensicherheit, falls das RAM ungepuffert ist. Die Daten im RAM gehen in diesem Fall verloren.

Speichern auf EPROM-Memory Card

Bausteine bzw. das Anwenderprogramm werden auf eine EPROM-Memory Card gespeichert, die dann in einen Schacht der CPU gesteckt wird.

Memory Cards sind portable Datenträger. Sie werden auf dem Erstellsystem beschrieben und dann in den dafür vorgesehenen Schacht der CPU eingesetzt

Die gespeicherten Daten bleiben bei Spannungsausfall und beim Urlöschen der CPU erhalten. Der Inhalt des EPROMs wird nach Urlöschen der CPU und nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr wieder in den RAM-Bereich des Speichers der CPU kopiert.

Speichern im integrierten EPROM

Für die CPU 312 gibt es noch die Möglichkeit, den Inhalt des RAMs in das integrierte EPROM zu speichern. Die Daten im integrierten EPROM bleiben bei Spannungsausfall remanent. Der Inhalt des integrierten EPROMs wird nach Netzausfall mit ungepuffertem RAM-Speicher bei Spannungswiederkehr und nach Urlöschen der CPU wieder in den RAM-Bereich des Speichers der CPU kopiert.

17.1.5 Nachladen von Bausteinen in das Zielsystem

Sie können Bausteine, die im Ladespeicher (RAM) bzw. Arbeitsspeicher der CPU des S7-Zielsystems vorhanden sind, mit einer neuen Version überschreiben (nachladen). Dabei wird der bis dahin vorhandene Stand überschrieben.

Die Vorgehensweise beim Nachladen entspricht der des Ladens von S7-Bausteinen. Es erfolgt lediglich eine Abfrage, ob der vorhandene Baustein überschrieben werden soll.

Ein im EPROM gespeicherter Baustein kann nicht gelöscht werden, wird aber bei der Nachladefunktion für ungültig erklärt. Der ihn ersetzende Baustein wird in das RAM geladen. Dabei entstehen im Lade- oder Arbeitsspeicher Lücken. Falls diese Lücken dazu führen, daß keine neuen Bausteine mehr ladbar sind, sollten Sie die Speicher komprimieren.

Hinweis

Im Fall eines Netzausfalls mit ungepuffertem RAM-Speicher und Spannungswiederkehr oder beim Urlöschen der CPU werden wieder die "alten" Bausteine aus dem EPROM gültig und geladen!

17.1.6 Laden über EPROM-Memory Cards

Voraussetzungen

Für Zugriffe auf EPROM-Memory Cards im Erstellsystem, die für ein S7-Zielsystem bestimmt sind, benötigen Sie die entsprechenden EPROM-Treiber. Für Zugriffe auf EPROM-Memory Cards, die für ein M7-Zielsystem bestimmt sind, muß das Flash File System installiert sein (nur bei PG 720/740/760 möglich). EPROM-Treiber und Flash File System werden bei der Installation des Basispakets von STEP 7 als Option angeboten. Wenn Sie einen PC einsetzen, so ist zum Speichern auf EPROM-Memory Cards zusätzlich ein externer Prommer erforderlich.

Sie können die Treiber auch nachträglich installieren. Rufen Sie dazu das entsprechende Dialogfeld über die Startleiste (**Start > Simatic > STEP 7 > Memory-Card parametrieren**) oder über die Systemsteuerung (Doppelklick auf das Symbol "Memory Card parametrieren") auf.

Speichern auf der Memory Card

Um Bausteine oder Anwenderprogramme in einer Memory Card zu speichern, gehen Sie wie folgt vor:

1. Stecken Sie die Memory Card in den Schacht Ihres Erstellsystems.
2. Öffnen Sie das Fenster "S7-Memory Card" wie folgt:
   - Klicken Sie auf das Symbol für "S7-Memory Card" in der Funktionsleiste. Aktivieren Sie ggf. die Anzeige dieser Leiste mit dem Menübefehl **Ansicht > Funktionsleiste**.
   - Wählen Sie alternativ den Menübefehl **Datei > S7-Memory Card > Öffnen**.
3. Öffnen bzw. aktivieren Sie ein Fenster, in dem Sie sich die zu speichernden Bausteine anzeigen lassen. Möglich sind:
   - Projektfenster, Ansicht "ONLINE"
   - Projektfenster, Ansicht "offline"
   - Bibliotheksfenster
   - Fenster "Erreichbare Teilnehmer"
4. Markieren Sie den Ordner "Bausteine" bzw. die Bausteine und kopieren Sie diese in das Fenster "S7-Memory Card".
5. Ist ein Baustein bereits auf der Memory Card vorhanden, wird eine Fehlermeldung ausgegeben. Löschen Sie in diesem Fall den Inhalt der Memory Card und wiederholen Sie die Schritte ab 2.

17.1.7 Hardware-Konfiguration und Verbindungsprojektierung separat laden

17.1.7.1 Laden einer Konfiguration in ein Zielsystem

Tip

Vor dem Laden sollten Sie mit dem Menübefehl **Station > Konsistenz prüfen** die Fehlerfreiheit Ihrer Stationskonfiguration prüfen. STEP 7 prüft dann, ob aus der aktuellen Konfiguration ladbare Systemdaten erzeugt werden können. Bei der Konsistenzprüfung werden Ihnen vorliegende Fehler in einem Fenster angezeigt.

Voraussetzungen zum Laden

- Das Programmiergerät ist über ein MPI-Kabel an die MPI-Schnittstelle der CPU angeschlossen
- Bei vernetzter Anlage (Programmiergerät ist an ein Subnetz angeschlossen): Sämtliche Baugruppen eines Subnetzes müssen unterschiedliche Teilnehmeradressen besitzen und der tatsächliche Aufbau muß mit der erstellten Netzkonfiguration übereinstimmen
- Die erstellte Konfiguration entspricht dem tatsächlichen Stationsaufbau. Eine Konfiguration kann nur dann in die Station geladen werden, wenn sie konsistent und fehlerfrei ist. Nur dann können Systemdatenbausteine (SDBs) erzeugt werden, die wiederum in die Baugruppen geladen werden
- Falls der Stationsaufbau Baugruppen enthält, die mit Optionspaketen konfiguriert und parametriert wurden: Das Optionspaket muß mit Autorisierung installiert sein

Vorgehensweise

- Wählen Sie den Menübefehl **Zielsystem > Laden in Baugruppe**
  STEP 7 führt Sie dann über Dialogfelder zum Ergebnis.

Die Konfiguration des gesamten Automatisierungssystems wird in die CPU geladen. CPU-Parameter werden sofort wirksam, die Parameter für die übrigen Baugruppen werden im Anlauf an die Baugruppen übertragen.

Hinweis

Teilkonfigurationen, z. B. die Konfiguration einzelner Baugruppenträger, können nicht in eine Station geladen werden. Aus Konsistenzgründen lädt STEP 7 immer die komplette Konfiguration in die Station.

Betriebszustand der CPU beim Laden ändern

Wenn Sie die Funktion **Zielsystem > Laden in Baugruppe** anstoßen, können Sie dialoggeführt folgende Aktionen per PG ausführen:

- CPU in STOP versetzen
  (falls Betriebsartenschalter auf RUN-P steht oder die Verbindung zur CPU durch Paßwort legitimiert wurde)
- Speicher komprimieren
  (falls nicht genügend zusammenhängender freier Speicher zur Verfügung steht)
- CPU wieder in RUN versetzen

17.1.7.2 Erstmaliges Laden der Netzkonfiguration

Vor dem erstmaligen Laden haben die am Subnetz angeschlossenen Baugruppen noch nicht ihre projektierte Teilnehmeradresse, sondern eine voreingestellte Adresse. Damit Ihr Netz einwandfrei arbeitet, müssen sämtliche Teilnehmer eines Subnetzes unterschiedliche Teilnehmeradressen besitzen.

- MPI-Subnetz mit Anschluß über die CPU
  CPUs werden mit der Default-Adresse 2 ausgeliefert. Da Sie diese Teilnehmeradresse nur einmal benutzen dürfen, müssen Sie die voreingestellte Teilnehmeradresse in allen anderen CPUs ändern.
- PROFIBUS- und Industrial Ethernet-Subnetze mit CPs
  Die CPs der Stationen, die über diese Subnetze betrieben werden, müssen projektiert und mit Teilnehmeradressen versorgt werden. Diese Adreßvergabe müssen Sie immer über die MPI-Schnittstelle der Station vornehmen, bevor Lade- und Kommunikationsvorgänge über das Subnetz erfolgen können (weitere Informationen hierzu siehe Handbücher SIMATIC NET, NCM S7 für PROFIBUS und NCM für Industrial Ethernet).

Wenn der Netzteilnehmer keine S7-Station ist...

Wenn der Netzteilnehmer keine S7-Station ist, müssen Sie die Netz- und Teilnehmereigenschaften mit dem dafür vorgesehenen Tool bzw. mit den dafür vorgesehenen Schaltern vorgeben. Das ist z. B. der Fall bei DP-Slaves, deren PROFIBUS-Adresse über Schalter eingestellt werden muß.

Stellen Sie sicher, daß diese Einstellungen mit den Einstellungen bei den Objekten in der Netzansicht übereinstimmt (PG/PC, Andere Station, S5-Station).

Ändern der PROFIBUS-Adresse bei DP-Slaves

An PROFIBUS-Subnetz angeschlossene DP-Slaves müssen ebenfalls eine eindeutige PROFIBUS-Adresse haben. Falls der anzuschließende DP-Slave die Funktion "Set_Slave_Add" unterstützt (z. B. ET 200C), können Sie die Adresse mit STEP 7 zuweisen:

Im SIMATIC Manager und beim Hardware konfigurieren können Sie mit dem Menübefehl **Zielsystem > PROFIBUS-Adresse vergeben** eine neue PROFIBUS-Adresse zuweisen.

Tip: Falls Sie die aktuelle Adreßvergabe nicht zweifelsfrei feststeht, sollten Sie die DP-Slaves einzeln an das PG/PC anschließen und umadressieren.

Ändern der Teilnehmeradresse bei S7-Stationen

Um die voreingestellte Teilnehmeradresse zu ändern, gehen Sie bei S7-Stationen folgendermaßen vor:

1. Konfigurieren Sie die Station; stellen Sie im Register "Allgemein" die Teilnehmeradresse (Schaltfäche "Eigenschaften" unter "Schnittstelle") der angeschlossenen Baugruppe ein(z. B. CPU).
2. Schalten Sie die Baugruppe in den Betriebszustand STOP und verbinden Sie Ihr Programmiergerät über eine Steckleitung mit der Schnittstelle der Baugruppe.
3. Ermitteln Sie die voreingestellte Teilnehmeradresse der angeschlossenen Baugruppe (z. B. über Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** im SIMATIC Manager).
4. Laden Sie die Konfiguration mit der neuen Teilnehmeradresse in das Zielsystem (d. h. in die angeschlossene Baugruppe):
   - In der Stationssicht (Hardware konfigurieren) mit dem Menübefehl **Zielsystem > Laden in Baugruppe**
   - In der Netzansicht (NetPro) markieren Sie die zu ladende Station und wählen den Menübefehl **Zielsystem > Laden > Markiere Stationen**. Geben Sie die ermittelte "alte" (zu diesem Zeitpunkt noch gültige) Adresse an!

17.1.7.3 Laden der Netzkonfiguration in ein Zielsystem

Voraussetzung

Im folgenden gehen wir davon aus, daß das komplette Projekt bereits projektiert wurde; d. h. Sie haben:

- Alle Stationen konfiguriert
- Alle Subnetze angelegt und deren Eigenschaften festgelegt
- Verbindungen projektiert (falls erforderlich)
- PG/PC-Schnittstelle eingestellt, so daß eine Kommunikation zwischen PG/PC und Automatisierungssystem über das angeschlossene Subnetz möglich ist.
- Die Projektierung auf Konsistenz geprüft

Erst dann, wenn eine Konfiguration fehlerfrei ist, d. h., wenn sämtliche vernetzte Baugruppen eines Subnetzes unterschiedliche Teilnehmeradressen besitzen und Ihr tatsächlicher Aufbau mit der erstellten Netzkonfiguration übereinstimmt, sollten Sie die Konfiguration über das Subnetz (PROFIBUS oder MPI) in die Zielsysteme laden.

17.1.7.4 Laden von Änderungen der Netzkonfiguration

Voraussetzungen

Sämtliche vernetzte Baugruppen eines Subnetzes haben unterschiedliche Teilnehmeradressen und der tatsächliche Aufbau stimmt mit der erstellten Netzkonfiguration überein.

Falls Sie eine neue Station an ein Subnetz anschließen und die voreingestellte Teilnehmeradresse im Subnetz bereits vorhanden ist, müssen Sie vorgehen wie im Abschnitt "Erstmaliges Laden" beschrieben.

Was wird wohin geladen?

Nach Übersetzen der Netzprojektierung (Menübefehl **Netz > Speichern und übersetzen**) bzw. nach **Zielsystem > Laden > ...** erzeugt NetPro Systemdatenbausteine (SDBs) für Baugruppen, die die Informationen in den SDBs interpretieren können. Die SDBs können Verbindungstabellen, Teilnehmeradressen, Subnetz-Eigenschaften, Ein-/Ausgangsadressen und Baugruppen-Parametrierungen enthalten.

Je nachdem, welchen Menübefehl Sie für das Laden wählen, werden unterschiedliche Inhalte bzw. unterschiedlich Zielsysteme geladen.

Hinweis

Nur bei der Option **Laden > Verbindungen und Netzübergänge** können Sie die beteiligten CPUs im Betriebszustand RUN-P laden. Bei allen anderen Optionen muß die CPU in den Betriebszustand STOP gebracht werden.

Menübefehl Zielsystem > Laden >	**Was wird geladen?**	**Wohin?**
... Markierte Stationen	Verbindungstabellen, Teilnehmeradressen, Subnetz-Eigenschaften, Ein-/Ausgangsadressen und Baugruppen-Parametrierungen der markierten Stationen	In die markierten Stationen
... Markierte und Partnerstationen	Verbindungstabellen, Teilnehmeradressen, Subnetz-Eigenschaften, Ein-/Ausgangsadressen und Baugruppen-Parametrierungen der markierten Station und der Verbindungspartner der markierten Station	In die markierte Station und in alle Stationen, die Verbindungspartner dieser Station sind
... Stationen am Subnetz	Verbindungstabellen, Teilnehmeradressen, Subnetz-Eigenschaften, Ein-/Ausgangsadressen und Baugruppen-Parametrierungen	Nacheinander in alle Stationen des markierten Subnetzes
... Markierte Verbindungen	Markierte Verbindungen (Mehrfachauswahl möglich)	In die lokale Station und (bei zweiseitigen Verbindungen) in die entsprechenden Partnerstationen
... Verbindungen und Netzübergänge	Verbindungen (möglich ist auch eine leere Verbindungstabelle) und Netzübergangs-Information	In die markierte Baugruppe (Im Betriebszustand RUN-P möglich!)

Vorgehensweise

1. Verbinden Sie das PG mit dem Subnetz, an das auch der zu ladende Teilnehmer angeschlossen ist.
2. Öffnen Sie NetPro.
3. Markieren Sie die zu ladende Station oder das Subnetz (bei..**Laden > Markiertes Subnetz**) in der Netzansicht.
4. Wählen Sie eine der oben beschrieben Optionen des Menübefehls **Zielsystem > Laden**.

Weitere Informationen

Weitere Informationen zu den Menübefehlen, die das Laden betreffen, finden Sie in der Hilfe zum Kontext (Menübefehl markieren und Taste F1 drücken).

17.1.7.5 Laden der Globaldaten-Konfiguration

Beim Übersetzen werden die Daten der GD-Tabelle in Systemdaten umgewandelt. Werden nach dem Übersetzen keine Fehler angezeigt, können Sie die Systemdaten in die CPUs übertragen:

- Wählen Sie den Menübefehl **Zielsystem > Laden**.

17.2 Laden aus dem Zielsystem in das PG

17.2.1 Laden aus dem Zielsystem in das PG

Diese Funktion unterstützt Sie bei folgenden Tätigkeiten:

- Sichern von Informationen aus dem Zielsystem (z.B. für Service-Zwecke)
- Schnelles Konfigurieren und Bearbeiten einer Station, wenn die Hardware-Komponenten zum Beginn der Konfigurierungstätigkeit vorhanden sind.

Sichern von Informationen aus dem Zielsystem

Diese Maßnahme kann erforderlich sein, wenn z.B. die Offline-Projektdaten des in der CPU laufenden Standes nicht oder nur teilweise vorhanden sind. Sie können sich dann zumindest den online verfügbaren Teil der Projektdaten auf Ihr PG holen.

Schnelles Konfigurieren

Sie können sich die Eingabe des Stationsaufbaus erleichtern, indem Sie nach Aufbau der Hardware und Neustart der Station die Konfigurationsdaten aus dem Zielsystem auf Ihr PG laden. Sie erhalten damit den Stationsaufbau mit den Typangaben der einzelnen Baugruppen. Danach müssen nur noch die einzelnen Baugruppen genauer spezifiziert (MLFB-Nr.) und parametriert werden.

Folgende Information wird in das PG geladen:

- S7-300: Konfiguration für den Zentralbaugruppenträger und eventuell vorhandene Erweiterungsbaugruppenträger
- S7-400: Konfiguration des Zentralbaugruppenträgers mit einer CPU und Signalbaugruppen ohne Erweiterungsbaugruppenträger
- Konfigurationsdaten zur dezentralen Peripherie können nicht in das PG geladen werden.

Dieser Informationsumfang wird geladen, wenn noch keine Projektierinformation auf dem Zielsystem vorliegt z.B. bei urgelöschten Systemen. Ansonsten liefert das "Laden in das PG" weit bessere Ergebnisse.

Bei S7-300-Systemen ohne dezentrale Peripherie brauchen Sie somit nur noch die Baugruppen genau zu spezifizieren (MLFB-Nr.) und zu parametrieren.

Hinweis

Beim Laden in das PG (ohne daß eine Offline-Konfiguration vorhanden ist) kann STEP 7 nicht alle Bestellnummern der Komponenten vollständig ermitteln.

Die "unvollständig" erscheinenden Bestellnummern können Sie beim Konfigurieren der Hardware mit dem Menübefehl **Extras > Baugruppe spezifizieren** eingeben. Auf diese Weise können Baugruppen, die STEP 7 nicht bekannt sind (d. h. nicht im Fenster “Hardware Katalog” erscheinen), parametriert werden; allerdings ohne daß Parameterregeln überwacht werden!

Einschränkungen beim Laden aus dem Zielsystem

Für die aus dem Zielsystem in das PG geladene Daten gelten folgende Einschränkungen:

- Bausteine enthalten keine symbolischen Namen für Parameter, Variablen und Marken.
- Bausteine enthalten keine Kommentare.
- Es wird das gesamte Programm mit allen Systemdaten ("AG-Abzug") auf das PG geladen. Dabei kann von den Systemdaten nur der Teil von "HW-Konfigurieren" wie gewohnt weiterbearbeitet werden.
- Die Daten zu den Themen "Globaldaten-Kommunikation (GD)", "Projektierung von symbolbezogenen Meldungen" können nicht weiterbearbeitet werden.
- Ein Forceauftrag wird nicht in das PG mitgeladen. Er muß über die Anzeige des Forceautrags separat als VAT gespeichert werden.
- Kommentare in den Dialogen der Baugruppen werden nicht geladen.
- Die Namen der Baugruppen werden nur dann angezeigt, wenn dies beim Konfigurieren angewählt wurde (HW-Konfig: Extras>Einstellungen, Objektnamen im Zielsystem speichern)

17.2.2 Station laden in PG

Mit dem Menübefehl **Zielsystem > Station laden in PG** können Sie die aktuelle Konfiguration und alle Bausteine aus dem zu wählenden Automatisierungssystem in das PG laden.

Dazu legt STEP 7 eine neue Station im aktuellen Projekt an, unter der die Konfiguration abgespeichert wird. Sie können den voreingestellten Namen der eingefügten Station (z.B. "SIMATIC 300-Station(1)") ändern. Die eingefügte Station wird sowohl in der Ansicht "online" als auch in der Ansicht "offline" angezeigt.

Der Menübefehl ist wählbar, wenn ein Projekt geöffnet ist. Die Markierung eines Objekts im Projektfenster oder die Ansicht (online oder offline) ist für den Menübefehl ohne Bedeutung.

Sie können diese Funktion benutzen, um sich das Konfigurieren zu erleichtern.

- Bei Zielsystemen S7-300 wird die Konfiguration für den vorhandenen Aufbau einschließlich Erweiterungsbaugruppenträger ohne dezentrale Peripherie (DP) geladen.
- Bei S7-400-Zielsystemen wird die Konfiguration des Baugruppenträgers ohne Erweiterungsbaugruppenträger und ohne dezentrale Peripherie geladen.

Bei S7-300-Systemen ohne dezentrale Peripherie brauchen Sie somit nur noch die Baugruppen genau zu spezifizieren (MLFB-Nr.) und zu parametrieren.

Einschränkungen bei Station laden in PG

Für die in das PG geladene Daten gelten folgende Einschränkungen:

- Bausteine enthalten keine symbolischen Namen für Formalparameter, temporäre Variablen und Marken.

- Bausteine enthalten keine Kommentare.
- Es wird das gesamte Programm mit allen Systemdaten ("AG-Abzug") auf das PG geladen, davon können aber nicht alle Daten sinnvoll weiterbearbeitet werden.
- Die Daten zu den Themen "Globaldaten-Kommunikation (GD)", "Projektierung von symbolbezogenen Meldungen", "Netzprojektierung" können nicht weiterbearbeitet werden.
- Forceaufträge können nicht auf das PG und von dort wieder auf das Zielsystem geladen werden.

17.2.3 Zurückladen von Bausteinen aus S7-CPU

Sie können mit dem SIMATIC Manager S7-Bausteine aus der CPU auf die Festplatte des Erstellsystems zurückladen. Die Möglichkeit des Ladens von Bausteinen in das PG nutzen:

- eine Sicherung des aktuellen Anwenderprogramms, das in der CPU geladen ist. Diese Sicherung kann dann z.B. im Servicefall nach evtl. Urlöschen der CPU vom Wartungspersonal übertragen werden.
- Sie können das Anwenderprogramm aus der CPU in das Erstellsystem laden und dort bearbeiten, z.B. im Servicefall zur Fehlersuche. Sie haben in diesem Fall keine Symbolik bzw. Kommentare zur Programmdokumentation zur Verfügung. Deswegen empfiehlt sich diese Vorgehensweise wirklich nur für den Servicefall.

17.2.4 Bearbeiten von geladenen Bausteinen im PG/PC

Die Möglichkeit zum Bearbeiten von Bausteinen im PG bietet Ihnen folgenden Nutzen:

- Sie können in der Testphase einen Baustein direkt in der CPU korrigieren und das Ergebnis dokumentieren.
- Sie können sich über die Ladefunktion aktuelle Inhalte von Bausteinen aus dem RAM-Ladespeicher der CPU in Ihr PG laden.

Hinweis

Zeitstempelkonflikt bei Arbeiten online und offline

Die anschließend beschriebenen Vorgehensweisen führen zu Zeitstempelkonflikten und sollten deshalb vermieden werden.

Zeitstempelkonflikte entstehen beim Online-Öffnen eines Bausteins, wenn

- online durchgeführte Änderungen nicht in das S7-Anwenderprogramm offline gespeichert wurden
- offline durchgeführte Änderungen nicht in die CPU geladen wurden.

Zeitstempelkonflikte entstehen beim Offline-Öffnen eines Bausteins, wenn

- ein Online-Baustein mit Zeitstempelkonflikt in das S7-Anwenderprogramm offline kopiert und anschließend der Baustein offline geöffnet wird.

Fallunterscheidung

Beim Laden von Bausteinen aus der CPU in das Programmiergerät müssen Sie zwei Fälle unterscheiden:

1. Das Anwenderprogramm, zu dem die Bausteine gehören, befindet sich im Programmiergerät.
2. Das Anwenderprogramm, zu dem die Bausteine gehören, befindet sich nicht im Programmiergerät.

Dieses hat zur Folge, daß Programmteile, die nicht in die CPU geladen werden können, nicht zur Verfügung stehen. Solche Programmteile sind:

- die Symboltabelle mit den symbolischen Namen für die Operanden sowie Kommentare
- Netzwerk-Kommentare eines FUP- oder KOP-Programms
- Zeilenkommentare eines AWL-Programms
- anwenderdefinierte Datentypen

17.2.5 Hardware-Konfiguration und Verbindungsprojektierung zurückladen

17.2.5.1 Zurückladen einer Konfiguration aus einer Station

Voraussetzung

Sie haben das Programmiergerät über ein MPI-Kabel an die MPI-Schnittstelle der CPU angeschlossen.

Tips

Laden Sie Stationen in ein neu erstelltes, leeres Projekt.

Stationen, die in besonderer Weise von anderen Stationen abhängen (I-Slave an einer DP-Master- Station, Empfänger- und Sender einer Konfiguration mit Direktem Datenaustausch/Querverkehr) sollten immer zusammen in ein Projekt geladen werden. Grund: Ohne das jeweilige "Pendant" einer solchen Station bleibt das Projekt inkonsistent!

Vorgehensweise

1. Wählen Sie den Menübefehl **Zielsystem > Laden in PG**.
   Es erscheint das Dialogfeld zum Öffnen der Konfiguration.
2. Wählen Sie das Projekt, in dem die Konfiguration später abgespeichert werden soll und quittieren Sie mit "OK".
3. Im anschließend erscheinenden Dialogfeld stellen Sie Teilnehmeradresse, Nr. des Baugruppenträgers und Steckplatz der Baugruppe ein, aus der die Konfiguration gelesen werden soll (i. d. R. eine CPU). Quittieren Sie mit "OK".

Dieser Konfiguration können Sie mit dem Menübefehl **Station > Eigenschaften** einen Stationsnamen geben und sie anschließend im voreingestellten Projekt abspeichern (Menübefehl **Station > Speichern**).

17.2.5.2 Zurückladen einer Netzkonfiguration (Laden in PG)

Einführung

Sie haben die Möglichkeit, den realen Netzaufbau Ihres Projektes Station für Station in Ihr PG zu laden.

Zum einen können Sie im SIMATIC Manager Station für Station die gesamte Konfiguration für ein Projekt in das PG laden (Menübefehl **Zielsystem > Laden in PG**). Daraufhin legt STEP 7 für jede zu ladende Station ein neues Stationsobjekt im aktuellen Projekt an.

Außerdem haben Sie beim Konfigurieren der Hardware die Möglichkeit, eine Stationskonfiguration zu laden (Menübefehl **Zielsystem > Laden in PG**).

Im folgendem wird gezeigt, wie Sie in NetPro die gesamte Netzkonfiguration Station für Station laden können.

Voraussetzungen

Das PG/PC ist am selben Subnetz angeschlossen wie die zu ladenden Stationen oder die Stationen sind über Netzübergänge erreichbar. Teilnehmeradressen und Baugruppenträger/Steckplätze der am Subnetz angeschlossenen Baugruppen sind bekannt.

Vorgehensweise

1. Verbinden Sie das PG mit dem Subnetz, an das auch der zu ladende Teilnehmer angeschlossen ist
2. Legen Sie ggf. ein neues Projekt für die geladene Netzkonfiguration an.
3. Öffnen Sie NetPro über ein Projekt, in das Sie die geladene Netzkonfiguration später speichern wollen (z. B. über ein neu angelegtes Projekt).
4. Wählen Sie den Menübefehl **Zielsystem > Station laden in PG**
   Der Menübefehl ist nur wählbar, wenn ein Projekt geöffnet ist
5. Geben Sie im Folgedialog über Teilnehmeradresse und Baugruppenträger/Steckplatz die Station an, die geladen werden soll.
   Das Objekt "Station" erscheint in der Netzansicht mit allen Baugruppen, die einen Netzanschluß haben. Die Subnetze, an die die Station angeschlossen ist, werden ebenfalls angezeigt. Den vom System vorgegebenen Namen der Station können Sie über den Menübefehl **Bearbeiten > Objekteigenschaften** ändern.
   Die projektierten Verbindungen werden ebenfalls geladen und sind sichtbar, wenn Sie eine Baugruppe, die Endpunkt von Verbindungen ist, markieren.
6. Sie können die Stationskonfiguration oder auch die Verbindungen ändern und anschließend die Änderungen in die Station laden. Bei Verbindungen, die mit Optionspaketen erstellt wurden, muß das Optionspaket installiert sein, damit diese Verbindungen bearbeitet und wieder in die Station geladen werden können.
7. Gehen Sie vor wie oben beschrieben, bis Sie alle gewünschten Stationen geladen haben.
8. Falls gewünscht, können Sie die Netzkonfiguration im aktuellen Projekt speichern (Menübefehl **Netz > Speichern** oder **..> Speichern und übersetzen**).

Besonderheiten von Verbindungen, die in das PG geladen wurden

In der Verbindungstabelle fehlt der offline projektierte Verbindungspartner - der Verbindungspartner ist "unspezifiziert". Adressierungsdetails sind aber über den Folgedialog zum Eigenschaftsdialog erreichbar.

Die Kommunikationsrichtung von PtP-Verbindungen kann nicht in allen Fällen von STEP 7 ermittelt werden; es wird aber gemeldet, welche Kommunikationrichtungen in Frage kommen.

17.2.6 Löschen im Zielsystem

17.2.6.1 Löschen des Lade-/Arbeitsspeichers und Urlöschen der CPU

Vor dem Laden Ihres Anwenderprogramms in das S7-Zielsystem, sollten Sie die CPU urlöschen, um sicherzustellen, daß sich keine "alten" Bausteine mehr auf der CPU befinden.

Voraussetzung für das Urlöschen

Zum Urlöschen muß sich die CPU im Betriebszustand STOP befinden (Betriebsartenschalter auf STOP oder Betriebsartenschalter auf RUN-P und Ändern des Betriebszustands nach STOP über den Menübefehl **Zielsystem > Betriebszustand**).

Urlöschen von S7-CPUs

Beim Urlöschen von S7-CPUs wird folgendes durchgeführt:

- Die CPU wird zurückgesetzt.
- Alle Anwenderdaten werden gelöscht (Bausteine und SDBs mit Ausnahme der MPI-Parametrierung)
- Die CPU bricht alle bestehenden Verbindungen ab.
- Wenn Daten auf einem EPROM vorhanden sind (Memory Card oder integriertes EPROM), kopiert die CPU nach dem Urlöschen den EPROM-Inhalt wieder in den RAM-Bereich des Speichers.

Der Inhalt des Diagnosepuffers und die Parameter der MPI-Schnittstelle bleiben erhalten.

Urlöschen von M7-CPUs/-FMs

Beim Urlöschen von M7-CPUs/-FMs wird folgendes durchgeführt:

- Der ursprüngliche Zustand wird hergestellt.
- Die SDBs mit Ausnahme der MPI-Parametrierung werden gelöscht.
- Die CPU/FM bricht alle bestehenden Verbindungen ab. Anwenderprogramme bleiben erhalten und laufen nach dem Umschalten von STOP auf RUN weiter.

Mit der Funktion "Urlöschen" können Sie den ursprünglichen Zustand der M7-CPU/-FM nach schwerwiegenden Fehlern wiederherstellen, indem Sie die aktuellen Systemdatenbausteine (SDB) aus dem Arbeitsspeicher löschen und die im Festwertspeicher befindlichen SDBs neu laden. In manchen Fällen ist zusätzlich ein Neustart (Warmstart) des Betriebssystems erforderlich. Dafür müssen Sie den M7 mit dem Betriebsartenschalter urlöschen (in Position MRES). Ein Reset über den Betriebsartenschalter von SIMATIC M7-CPUs/FMs ist nur möglich, wenn dort das Betriebssystem RMOS32 verwendet wird.

17.2.6.2 Löschen von S7-Bausteinen auf dem Zielsystem

Das Löschen einzelner Bausteinen in der CPU kann während der Testphase des CPU-Programms notwendig werden. Bausteine sind im Anwenderspeicher der CPU entweder im EPROM oder RAM gespeichert (abhängig von CPU und Ladevorgang).

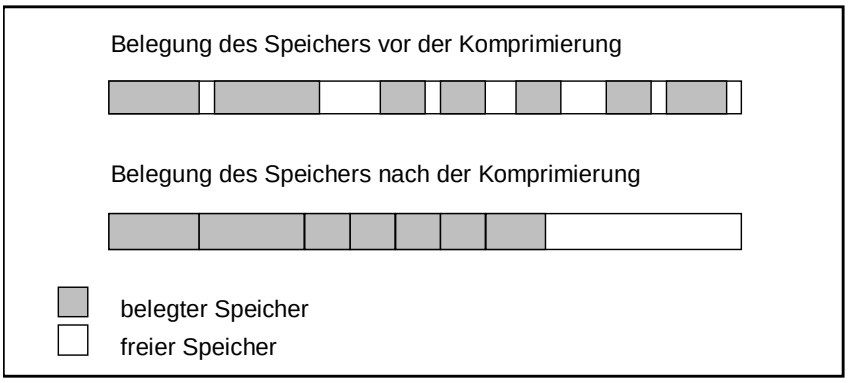
- Bausteine, die im RAM geladen sind, können Sie direkt löschen. Der belegte Speicherplatz wird im Ladespeicher und im Arbeitsspeicher freigegeben.
- Bausteine, die im integrierten EPROM gespeichert sind, werden immer nach dem Urlöschen der CPU in den RAM-Bereich kopiert. Die Kopien im RAM-Bereich können direkt gelöscht werden. Die gelöschten Bausteine werden dabei im EPROM bis zum nächsten Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher für ungültig markiert. Nach Urlöschen oder Netzausfall mit ungepuffertem RAM-Speicher werden die "gelöschten" Bausteine wieder vom EPROM in das RAM kopiert und aktiv. Bausteine im integrierten EPROM (z.B. bei CPU 312) werden durch Überschreiben mit dem neuen RAM-Inhalt gelöscht.
- EPROM-Memory Cards müssen im Erstellsystem gelöscht werden.

17.2.7 Komprimieren des Anwenderspeichers (RAM)

17.2.7.1 Entstehung von Lücken im Anwenderspeicher (RAM)

Durch mehrfaches Löschen und Nachladen von Bausteinen entstehen im Anwenderspeicher (Lade- und Arbeitsspeicher) Lücken, die den nutzbaren Speicherbereich verringern. Durch Komprimieren werden die vorhandenen Bausteine lückenlos im Anwenderspeicher angeordnet und es entsteht ein zusammenhängender freier Speicherbereich.

Nachfolgendes Bild zeigt schematisch, wie belegte Speicherblöcke durch die Funktion ”Speicher komprimieren” verschoben werden.


Belegung des Speichers vor der Komprimierung
Belegung des Speichers nach der Komprimierung
belegter Speicher
freier Speicher

Möglichst im Betriebszustand "STOP" komprimieren

Nur das Komprimieren im Betriebszustand "STOP" schließt alle Speicherlücken. Wenn Sie im Betriebszustand RUN-P (Stellung des Betriebsartenschalters) komprimieren, so werden die gerade bearbeiteten Bausteine nicht verschoben, da sie geöffnet sind. Im Betriebszustand RUN (Stellung des Betriebsartenschalters) kann die Funktion Komprimieren nicht durchgeführt werden (Schreibschutz!).

17.2.7.2 Komprimieren des Speicherinhalts einer S7-CPU

Möglichkeiten zum Komprimieren

Sie haben zwei Möglichkeiten zum Komprimieren des Anwenderspeichers:

- Tritt beim Laden ein Speicherplatzmangel auf dem Zielsystem auf, so wird ein Dialogfeld aufgeblendet, das Sie auf den Fehler hinweist. Durch Betätigen der entsprechenden Schaltfläche im Dialogfeld können Sie den Speicher komprimieren.
- Als vorbeugende Maßnahme können Sie sich die Speicherauslastung anzeigen lassen (Menübefehl **Zielsystem > Baugruppenzustand..../Register Speicher**) und ggf. die Komprimierung anstoßen.

Vorgehensweise

1. Markieren Sie das S7-Programm in der Ansicht online oder im Fenster "Erreichbare Teilnehmer".
2. Wählen Sie den Menübefehl **Zielsystem > Baugruppenzustand**.
3. Wählen Sie im folgenden Dialogfeld das Register "Speicher". In diesem Register gibt es eine Schaltfläche für das Komprimieren des Speichers, falls die CPU diese Funktion unterstützt.

18 Testen mit der Variablentabelle

18.1 Einführung zum Testen mit Variablentabellen

Variablentabellen bieten den Vorteil unterschiedliche Testumgebungen abzuspeichern zu können. So können Tests und Beobachtungen während einer Inbetriebsetzung oder zu Service- und Wartungszwecken mühelos reproduziert werden. Die Anzahl der abspeicherbaren Variablentabellen ist nicht begrenzt.

Beim Testen mit Variablentabellen stehen Ihnen folgende Funktionen zur Verfügung:

- **Beobachten von Variablen**
  Damit lassen sich die aktuellen Werte einzelner Variablen eines Anwenderprogramm bzw. einer CPU am PG/PC anzeigen.
- **Steuern von Variablen**
  Mit dieser Funktion können Sie einzelnen Variablen eines Anwenderprogramms bzw. einer CPU feste Werte zuweisen. Ein einmaliges und sofortiges Steuern ist auch beim Testen mit Programmstatus möglich.
- **PA freischalten und Steuerwerte aktivieren**
  Diese beiden Funktionen geben Ihnen die Möglichkeit, einzelnen Peripherieausgängen einer CPU im Betriebszustand STOP feste Werte zuzuweisen.
- **Forcen von Variablen**
  Verwenden Sie diese Funktion, um einzelnen Variablen eines Anwenderprogramms bzw. einer CPU mit einem Wert fest zu belegen, der vom Anwenderprogramm nicht überschrieben werden kann.

Sie können sich die Werte von folgenden Variablen vorgeben bzw. anzeigen lassen:

- Eingänge, Ausgänge, Merker, Zeiten und Zähler
- Inhalte von Datenbausteinen
- Peripherie

Die Variablen, die Sie anzeigen oder steuern wollen, fassen Sie in Variablentabellen zusammen.

Wann und wie oft die Variablen beobachtet bzw. mit Werten überschrieben werden, bestimmen Sie durch Definieren von Triggerpunkt und Triggerbedingung.

18.2 Prinzipielle Vorgehensweise beim Beobachten und Steuern mit Variablentabellen

Um die Funktionen **Beobachten** und **Steuern** zu benutzen, gehen Sie folgendermaßen vor:

1. Erzeugen Sie eine neue Variablentabelle oder öffnen Sie eine bereits vorhandene.
2. Bearbeiten bzw. überprüfen Sie die Variablentabelle.
3. Stellen Sie eine Online-Verbindung zwischen der aktuellen Variablentabelle und der gewünschten CPU mit dem Menübefehl **Zielsystem > Verbindung herstellen zu >...** her.
4. Wählen Sie mit dem Menübefehl **Variable > Trigger** einen geeigneten Triggerpunkt aus und stellen Sie die Triggerbedingung ein.
5. Die Menübefehle **Variable > Beobachten** und **Variable > Steuern** schalten die jeweilige Funktion ein und wieder aus.
6. Speichern Sie die fertig erstellte Variablentabelle mit dem Menübefehl **Tabelle > Speichern** oder **Tabelle > Speichern unter**, damit Sie diese jederzeit wieder aufrufen können.

18.3 Bearbeiten und Speichern von Variablentabellen

18.3.1 Erstellen und Öffnen einer Variablentabelle

Bevor Sie Variablen beobachten oder steuern können, müssen Sie eine Variablentabelle (VAT) erstellen und die gewünschten Variablen eintragen. Um eine Variablentabelle zu erstellen, haben Sie folgende Möglichkeiten:

Im SIMATIC Manager:

- Markieren Sie den Ordner ”Bausteine” und wählen Sie mit den Menübefehl **Einfügen > S7-Baustein > Variablentabelle**. Im Folgedialog können Sie die Tabelle benennen (Eingabefeld "Symbolischer Name"). Dieser Name wird im Projektfenster angezeigt. Die Variablentabelle können Sie durch Doppelklick auf das Objekt öffnen.
- Wählen Sie aus der Liste der erreichbaren Teilnehmer eine Verbindung oder in der Online-Ansicht ein S7-/M7-Programm aus. Mit dem Menübefehl **Zielsystem > Variable beobachten/steuern** erzeugen Sie eine unbenannte Variablentabelle.

In "Variable beobachten/steuern":

- Sie können mit dem Menübefehl **Tabelle > Neu** eine neue Variablentabelle erzeugen, die noch keinem S7/M7-Programm zugeordnet sind. Bereits vorhandene Tabellen öffnen Sie mit **Tabelle > Öffnen**.
- Sie können die entsprechenden Symbole in der Funktionsleiste zum Erzeugen oder Öffnen von Variablentabellen benutzen.

Eine einmal erstellte Variablentabelle können Sie speichern, ausdrucken und immer wieder zum Beobachten und Steuern verwenden.

18.3.2 Kopieren/Verschieben von Variablentabellen

Variablentabellen können Sie in Baustein-Ordner eines S7/M7-Programms kopieren oder verschieben.

Beim Kopieren und Verschieben von Variablentabellen müssen Sie folgendes beachten:

- In der Symboltabelle des Zielprogramms werden dort bereits vorhandene Symbole aktualisiert.
- Beim Verschieben einer Variablentabelle werden auch die entsprechenden Symbole aus der Symboltabelle des Quellprogramms in die Symboltabelle des Zielprogramms verschoben.
- Beim Löschen von Variablentabellen aus dem Baustein-Ordner werden auch die entsprechenden Symbole aus der Symboltabelle des S7/M7-Programms gelöscht.
- Ist im Zielprogramm bereits eine Variablentabelle mit demselben Namen vorhanden, erhalten Sie beim Kopieren die Möglichkeit die Variablentabelle umzubennen (defaultmäßig wird zum bestehenden Namen eine Nummer angehängt).

18.3.3 Speichern einer Variablentabelle

Gespeicherte Variablentabellen können Sie beim erneuten Test Ihres Programms wieder zum Beobachten und Steuern verwenden.

1. Speichern Sie die Variablentabelle mit dem Menübefehl **Tabelle > Speichern.**
2. Wurde die Variablentabelle neu erstellt, so müssen Sie nun einen Namen für die Variablentabelle angeben, z.B. "Programmtest_1".

Beim Speichern der Variablentabelle werden alle aktuellen Einstellungen sowie das Tabellenformat gespeichert, d.h. die Einstellungen im Menüpunkt Trigger werden gespeichert.

18.4 Eingeben von Variablen in Variablentabellen

18.4.1 Einfügen von Operanden oder Symbolen in eine Variablentabelle

Bestimmen Sie die Variablen, deren Werte Sie vorgeben oder beobachten wollen und tragen Sie diese in die Variablentabelle ein. Gehen Sie dabei von "außen" nach "innen" vor, d.h. wählen Sie zuerst die Eingänge und danach die Variablen, die von den Eingängen beeinflußt werden bzw. die die Ausgänge beeinflussen und zum Schluß die Ausgänge.

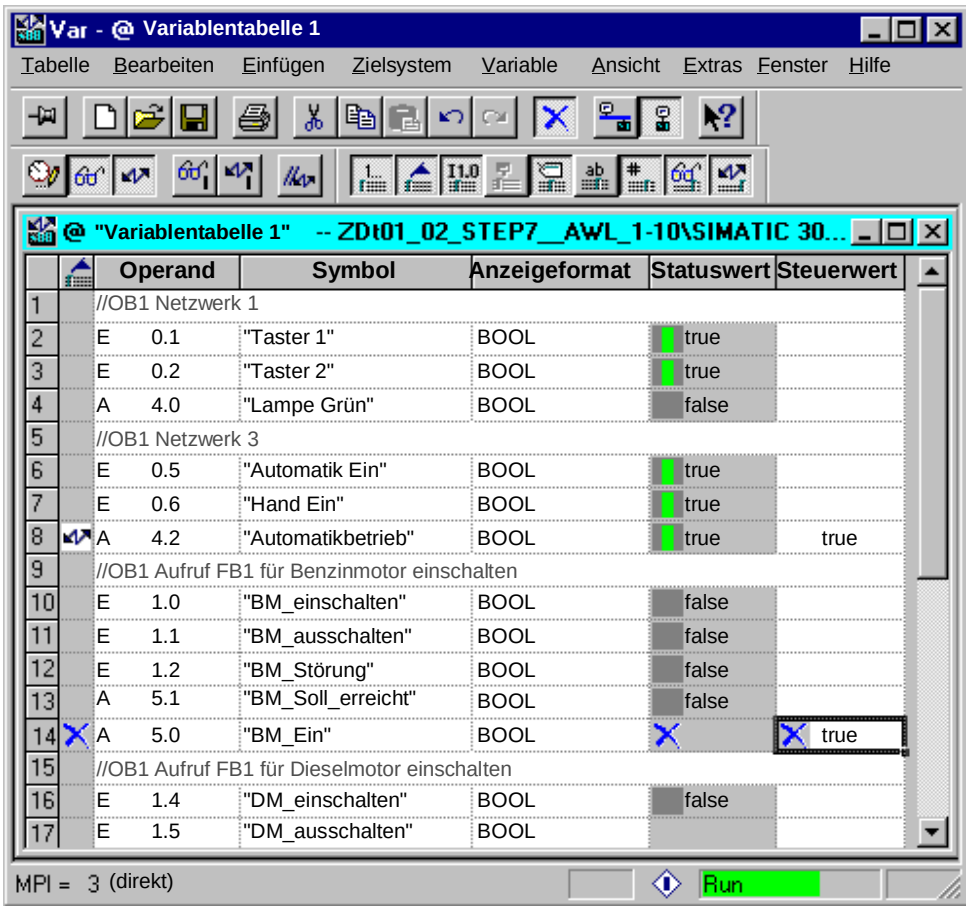
Wenn Sie z.B. das Eingangsbit 1.0, das Merkerwort 5 und das Ausgangsbyte 0 beobachten möchten, geben Sie in der Operandenspalte folgendes ein:

Beispiel:

E 1.0
MW 5
AB 0

Beispiel für eine ausgefüllte Variablentabelle

Nachfolgendes Bild zeigt eine Variablentabelle mit den eingeblendeten Spalten: Operand, Symbol, Anzeigeformat, Statuswert und Steuerwert


Var - @ Variablentabelle 1
Tabelle Bearbeiten Einfügen Zielsystem Variable Ansicht Extras Fenster Hilfe
@ "Variablentabelle 1" -- ZDt01_02_STEP7__AWL_1-10\SIMATIC 30...
MPI = 3 (direkt)
Run

		Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1		//OB1 Netzwerk 1				
2		E 0.1	"Taster 1"	BOOL	true	
3		E 0.2	"Taster 2"	BOOL	true	
4		A 4.0	"Lampe Grün"	BOOL	false	
5		//OB1 Netzwerk 3				
6		E 0.5	"Automatik Ein"	BOOL	true	
7		E 0.6	"Hand Ein"	BOOL	true	
8		A 4.2	"Automatikbetrieb"	BOOL	true	true
9		//OB1 Aufruf FB1 für Benzinmotor einschalten				
10		E 1.0	"BM_einschalten"	BOOL	false	
11		E 1.1	"BM_ausschalten"	BOOL	false	
12		E 1.2	"BM_Störung"	BOOL	false	
13		A 5.1	"BM_Soll_erreicht"	BOOL	false	
14		A 5.0	"BM_Ein"	BOOL		true
15		//OB1 Aufruf FB1 für Dieselmotor einschalten				
16		E 1.4	"DM_einschalten"	BOOL	false	
17		E 1.5	"DM_ausschalten"	BOOL		

Hinweise zum Einfügen von Symbolen

- Die zu steuernde Variable geben Sie mit ihrem Operand oder als Symbol an. Operanden und Symbole können Sie sowohl in die Spalte "Operand" als auch in die Spalte "Symbol" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
  Falls in der Symboltabelle ein entsprechendes Symbol definiert ist, wird der Eintrag in der Symbolspalte bzw. der Operandenspalte automatisch ergänzt.
- Sie können nur solche Symbole eintragen, die bereits in der Symboltabelle definiert sind.
- Ein Symbol muß genau so eingegeben werden, wie es in der Symboltabelle definiert ist.
- Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommas eingeschlossen werden (z.B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Um neue Symbole in der Symboltabelle zu definieren, wählen Sie den Menübefehl **Extras > Symboltabelle**. Symbole können auch aus der Symboltabelle kopiert und in eine Variablentabelle eingefügt werden.

Syntaxprüfung

Wenn Sie Variablen in die Variablentabelle eintragen, so wird vor Verlassen der Zeile eine Syntaxprüfung durchgeführt. Fehlerhafte Einträge werden rot gekennzeichnet. Wenn Sie den Cursor in eine rot markierte Zeile stellen, können Sie in der Statuszeile die Ursache des Fehlers ablesen. Mit der Taste F1 erhalten Sie Hinweise zur Behebung.

Maximale Größe

Eine Variablentabelle kann maximal 255 Zeichen pro Zeile umfassen. Ein Umbruch in die nächste Zeile ist nicht möglich. Ihre Länge ist auf 1024 Zeilen festgelegt. Damit ist die maximale Größe erreicht.

18.4.2 Einfügen von Steuerwerten

Steuerwert als Kommentar

Wenn Sie den "Steuerwert" einer Variablen unwirksam schalten wollen, so verwenden Sie den Menübefehl **Variable > Steuerwert als Kommentar.** Ein Kommentarzeichen ”//” vor dem zu steuernden Wert einer Variablen zeigt deren Unwirksamkeit an. Anstelle des Menübefehlaufrufs, kann dem "Steuerwert" auch das Kommentarzeichen ”//” vorangestellt werden. Die Unwirksamkeit des "Steuerwertes", kann durch den erneuten Aufruf des Menübefehls **Variable > Steuerwert als Kommentar** oder der Entfernung des Kommentarzeichens wieder aufgehoben werden.

18.4.3 Obergrenzen für die Eingabe von Zeiten

Beachten Sie folgende Obergrenzen für die Eingabe von Zeiten:

W#16#3999 (Maximalwert im BCD-Format)

Beispiele:

Operand		Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
T	1	SIMATIC_ZEIT	137	S5TIME#130MS	Konvertierung in Millisekunden
MW	4	SIMATIC_ZEIT	137	S5TIME#890MS	Darstellung im BCD-Format möglich
MW	4	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW	6	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat SIMATIC_ZEIT nicht wählbar

Hinweis

- Sie können Zeiten in Millisekunden-Schritten eingeben, aber der eingegebene Wert wird an ein Zeitraster angepaßt. Die Größe des Zeitrasters ist abhängig von der Größe des eingegebenen Zeitwertes (137 ergibt 130ms, die 7 ms wurden abgerundet).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl ! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als SIMATIC_ZEIT dargestellt werden kann, wird automatisch auf das voreingestellte Format (hier: HEX, siehe Anzeigeformat wählen, Voreinstellung Menü Ansicht) umgeschaltet, so daß der eingegebene Wert angezeigt werden kann.

BCD-Format für Variablen im Format SIMATIC_ZEIT

Werte von Variablen im Format SIMATIC_ZEIT werden im BCD-Format eingegeben.
Die 16 Bit haben folgende Bedeutung:

| 0 0 x x | h h h h | z z z z | e e e e |

Bit 15 und 14 sind immer Null.
Bit 13 und 12 (mit xx markiert) legen den Multiplikator für die Bits 0 bis 11 fest:

00 => Multiplikator 10 Millisekunden
01 => Multiplikator 100 Millisekunden
10 => Multiplikator 1 Sekunde
11 => Multiplikator 10 Sekunden

Bit 11 bis 8 Hunderter (hhhh)
Bit 7 bis 4 Zehner (zzzz)
Bit 3 bis 0 Einer (eeee)

18.4.4 Obergrenzen für die Eingabe von Zählern

Bitte beachten Sie folgende Obergrenzen für die Eingabe von Zählern:

Obergrenzen für Zähler: C#999
W#16#0999 (Maximalwert im BCD-Format)

Beispiele:

Operand		Anzeigeformat	Eingabe	Anzeige Steuerwert	Erläuterung
Z	1	ZÄHLER	137	C#137	Konvertierung
MW	4	ZÄHLER	137	C#89	Darstellung im BCD-Format möglich
MW	4	HEX	137	W#16#0089	Darstellung im BCD-Format möglich
MW	6	HEX	157	W#16#009D	Darstellung im BCD-Format nicht möglich, deshalb ist das Anzeigeformat ZÄHLER nicht wählbar

Hinweis

- Wenn Sie bei einem Zähler eine Dezimalzahl eingeben und den Wert nicht durch C# kennzeichnen, so wird dieser Wert automatisch in das BCD-Format umgerechnet (137 ergibt C#137).
- Die Steuerwerte von Operanden vom Datentyp WORD, z. B. EW 1, werden ins BCD-Format konvertiert. Nicht jedes Bitmuster ist jedoch eine gültige BCD-Zahl! Wenn bei einem Operanden vom Datentyp WORD die Eingabe nicht als ZÄHLER dargestellt werden kann, wird automatisch auf das voreingestellte Format (hier: HEX, siehe Anzeigeformat wählen, Voreinstellung (Menü Ansicht) umgeschaltet, so daß der eingegebene Wert angezeigt werden kann.

18.4.5 Einfügen von Kommentarzeilen

Kommentarzeilen werden durch das Kommentarzeichen "//" eingeleitet.

Wenn Sie eine oder mehrere Zeilen der Variablentabelle unwirksam schalten wollen, verwenden Sie den Menübefehl **Bearbeiten > Zeile nicht wirksam** oder das entsprechende Symbol in der Funktionsleiste.

18.5 Beispiele zur Eingabe in Variablentabellen

18.5.1 Beispiel für die Eingabe von Operanden in Variablentabellen

Erlaubter Operand:	Datentyp:	Beispiel (Mnemonik deutsch):
Eingang | Ausgang | Merker	BOOL	E 1.0 | A 1.7 | M 10.1
Eingang | Ausgang | Merker	BYTE	EB 1 | AB 10 | MB 100
Eingang | Ausgang | Merker	WORD	EW 1 | AW 10 | MW 100
Eingang | Ausgang | Merker	DWORD	ED 1 | AD 10 | MD 100
Peripherie (Eingang | Ausgang)	BYTE	PEB 0 | PAB 1
Peripherie (Eingang | Ausgang)	WORD	PEW 0 | PAW 1
Peripherie (Eingang | Ausgang)	DWORD	PED 0 | PAD 1
Zeiten	TIMER	T 1
Zähler	COUNTER	Z 1
Datenbaustein	BOOL	DB1.DBX 1.0
Datenbaustein	BYTE	DB1.DBB 1
Datenbaustein	WORD	DB1.DBW 1
Datenbaustein	DWORD	DB1.DBD 1

Hinweis:

Die Eingabe von "DB0. .." ist aufgrund interner Benutzung nicht erlaubt.

Im Fenster Forcewerte:

- Beim Forcen von S7-300-Baugruppen sind nur Eingänge, Ausgänge und Peripherie (Ausgänge) erlaubt.
- Beim Forcen von S7-400-Baugruppen sind nur Eingänge, Ausgänge, Merker und Peripherie (Eingänge/Ausgänge) erlaubt.

18.5.2 Beispiel für die Eingabe eines zusammenhängenden Operandenbereichs

Öffnen Sie eine Variablentabelle und lassen Sie sich das Dialogfeld "Bereich einfügen" mit dem Menübefehl **Einfügen > Bereich** anzeigen.

Bei den Angaben im Dialogfeld werden folgende Zeilen für Merker in die Variablentabelle eingefügt:

Ab Operand: M 3.0

Anzahl: 10

Anzeigeformat: BIN

Operand	Anzeigeformat
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

Beachten Sie in diesem Beispiel, wie sich die Bezeichnung in der Spalte "Operand" nach dem achten Eintrag ändert.

18.5.3 Beispiele für die Eingabe von Steuer-/Forcewerten

Bitoperanden

Mögliche Bitoperanden	erlaubte Steuer-/Forcewerte
E1.0	true
M1.7	false
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

Byteoperanden

mögliche Byteoperanden	erlaubte Steuer-/Forcewerten
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	12

Wortoperanden

mögliche Wortoperanden	erlaubte Steuer-/Forcewerte
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	12345
MW 3	12345
MW 5	S5t#12s340ms
MW 7	0.3s oder 0,3s
MW 9	C#123
MW 11	d#1990-12-31

Doppelwortoperanden

mögliche Doppelwortoperanden	erlaubte Steuer-/Forcewerten
ED 1	2#00110011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd'
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

Zeit

mögliche Operanden vom Typ Zeit	erlaubte Steuer-/Forcewerte	Erläuterung
T 1	0	Zeitwert in Millisekunden (ms)
T 12	20	Zeitwert in Millisekunden (ms)
T 14	12345	Zeitwert in Millisekunden (ms)
T 16	s5t#12s340ms	Zeitwert 12s 340ms
T 18	1.3	Zeitwert 1s 300 ms
T 20	1.3s	Zeitwert 1s 300 ms

Das Steuern einer Zeit beeinflußt nur den Wert, nicht den Zustand. D.h die Zeit T1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U T1 wird nicht verändert.

Die Zeichenfolgen "s5t", "s5time" können sowohl klein als auch groß geschrieben werden.

Zähler

Mögliche Operanden vom Typ Zähler	erlaubte Steuer-/Forcewerten
Z 1	0
Z 14	20
Z 16	c#123

Das Steuern eines Zählers beeinflußt nur den Wert, nicht den Zustand. D.h der Zähler Z1 kann auf den Wert 0 gesteuert werden, das Verknüpfungsergebnis bei U Z1 wird nicht verändert.

18.6 Herstellen einer Verbindung zur CPU

18.6.1 Herstellen einer Verbindung zur CPU

Damit Sie die Variablen, die sie in Ihrer aktuellen Variablentabelle (VAT) zusammengefaßt haben, beobachten oder steuern können, müssen Sie eine Verbindung zur entsprechenden CPU herstellen. Es ist möglich, jede Variablentabelle mit einer anderen CPU zu verbinden.

Anzeige der Online-Verbindung

Wenn eine Online-Verbindung besteht, so wird dies in der Titelzeile des Variablentabellenfensters mit dem Begriff ”ONLINE” angezeigt. In der Statuszeile werden, abhängig von der CPU, die Betriebszustände "RUN", "STOP", "GETRENNT" oder "VERBUNDEN" abgebildet.

Herstellen der Online-Verbindung zur CPU

Besteht keine Online-Verbindung zur gewünschten CPU, definieren Sie diese mit dem Menübefehl **Zielsystem > Verbindung herstellen zu >...** zur gewünschten CPU, um die Variablen zu beobachten oder zu steuern. Alternativ können Sie auch auf die entsprechenden Schaltflächen in der Funktionsleiste klicken:

Verbindung zur projektierten CPU

Verbindung zur direkt angeschlossenen CPU (z. B. MPI=2 (direkt))

Abbauen der Online-Verbindung zur CPU

Mit dem Menübefehl **Zielsystem > Verbindung abbauen** brechen Sie die Verbindung zwischen Variablentabelle und CPU wieder ab.

Hinweis

Wenn Sie eine unbenannte Variablentabelle mit dem Menübefehl **Tabelle > Neu** erzeugt haben, können Sie eine Verbindung zur letzten projektierten CPU herstellen, falls sie definiert ist.

18.7 Beobachten von Variablen

18.7.1 Einführung zum Beobachten von Variablen

Zum Beobachten von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion ”Beobachten” mit dem Menübefehl **Variable > Beobachten** ein. Die Werte der ausgewählten Variablen werden, abhängig vom eingestellten Triggerpunkt und der eingestellten Triggerbedingung, in der Variablentabelle angezeigt. Wenn Sie die Triggerbedingung ”Permanent” eingestellt haben, schalten Sie die Funktion ”Beobachten” mit dem Menübefehl **Variable > Beobachten** wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort mit dem Menübefehl **Variable > Statuswerte aktualisieren**. Die aktuellen Werte der ausgewählten Variablen werden in der Variablentabelle angezeigt.

Beobachten abbrechen mit ESC-Taste

Wenn Sie - bei laufendem "Beobachten" - die Taste ESC drücken, wird "Beobachten" ohne Rückfrage beendet.

18.7.2 Festlegen des Triggers zum Beobachten von Variablen

Sie können sich zum Beobachten die aktuellen Werte einzelner Variablen eines Anwenderprogramms an einem vorgegebenen Punkt im Ablauf des Programms (Triggerpunkt) am PG anzeigen lassen.

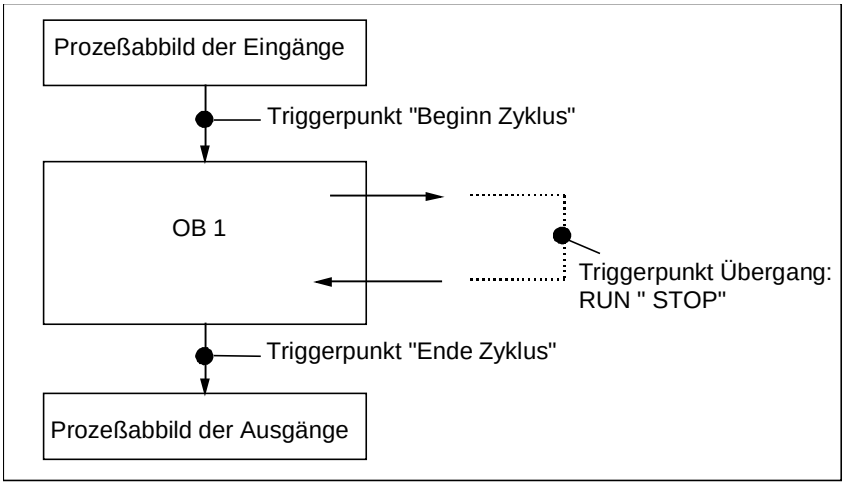
Mit der Wahl des Triggerpunktes bestimmen Sie den Zeitpunkt, an dem die Statuswerte von Variablen angezeigt werden.

Mit dem Menübefehl **Variable > Trigger** können Sie den Triggerpunkt und eine Triggerbedingung einstellen

Trigger	**Einstellungsmöglichkeiten**
Triggerpunkt	Beginn Zyklus Ende Zyklus Übergang von RUN nach STOP.
Triggerbedingung	einmalig permanent

Triggerpunkt

Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf ”Zyklusbeginn” und den Triggerpunkt für Steuern auf ”Zyklusende” einstellen!

Trigger sofort

Sie können die Werte ausgewählter Variablen mit dem Menübefehl **Variable > Statuswerte aktualisieren**. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktionen werden hauptsächlich im STOP-Zustand zum Steuern und Beobachten angewendet.

Triggerbedingung

Nachfolgende Tabelle zeigt, was die eingestellte Triggerbedingung für das Beobachten von Variablen bewirkt:

	Triggerbedingung Einmalig	**Triggerbedingung Permanent**
Beobachten von Variablen	*Einmaliges Aktualisieren* abhängig vom Triggerpunkt	*Beobachten mit definiertem Trigger* Beim Testen eines Bausteins können Sie die Weiterverarbeitung genau mitverfolgen.

18.8 Steuern von Variablen

18.8.1 Einführung zum Steuern von Variablen

Zum Steuern von Variablen haben Sie folgende Möglichkeiten:

- Schalten Sie die Funktion ”Steuern” mit dem Menübefehl **Variable > Steuern** ein. Das Anwenderprogramm übernimmt die Steuerwerte für die ausgewählten Variablen aus der Variablentabelle, abhängig vom jeweils eingestellten Triggerpunkt und der eingestellten Triggerbedingung. Wenn Sie die Triggerbedingung ”Permanent” eingestellt haben, schalten Sie die Funktion ”Steuern” mit dem Menübefehl **Variable > Steuern** wieder aus.
- Aktualisieren Sie die Werte der ausgewählten Variablen einmalig und sofort mit dem Menübefehl **Variable > Steuerwert aktivieren**.

Weitere Möglichkeiten bieten die Funktionen Forcen und Peripherieausgänge (PA) freischalten.

Beachten Sie beim Steuern:

- Es werden nur Operanden gesteuert, die zu Beginn des Steuerns in der Variablentabelle sichtbar waren.
  Wird der sichtbare Bereich der Variablentabelle nach Beginn des Steuerns verkleinert, werden eventuell Operanden gesteuert, welche nicht mehr sichtbar sind.
  Wird der sichtbare Bereich der Variablentabelle vergrößert, gibt es eventuell Operanden, die nicht gesteuert werden.
- Steuern kann nicht rückgängig gemacht werden (z. B. mit **Bearbeiten > Rückgängig**).
- Beim permanenten Steuern ist kein Scrollen des Bildschirms möglich.

Gefahr

Ein Verändern der Variablenwerte bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!
Vergewissern Sie sich, daß keine gefährlichen Zustände eintreten können, bevor Sie die Funktion “Steuern” ausführen!

Steuern abbrechen mit ESC-Taste

Wenn Sie - bei laufendem "Steuern" - die Taste ESC drücken, wird "Steuern" ohne Rückfrage beendet.

18.8.2 Festlegen des Triggers zum Steuern von Variablen

Sie können einzelnen Variablen eines Anwenderprogramms an einem vorgegebenen Punkt im Ablauf des Programms (Triggerpunkt) einmalig oder permanent feste Werte zuweisen.

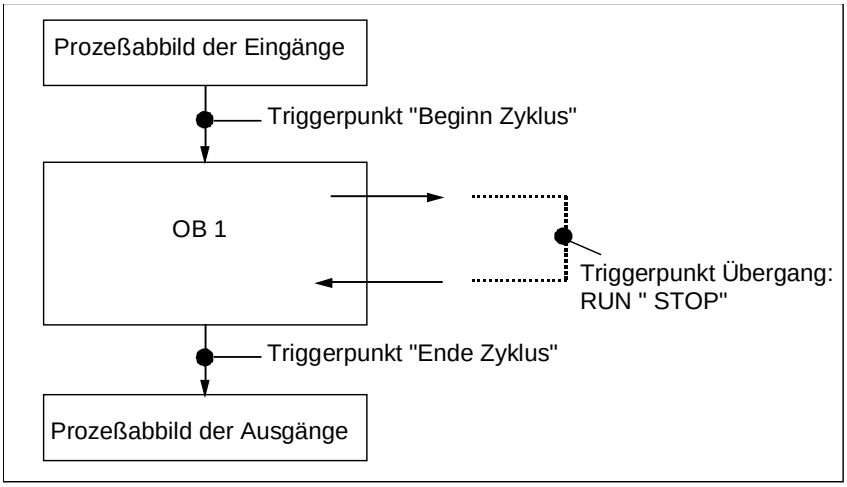
Mit der Wahl des Triggerpunktes bestimmen Sie den Zeitpunkt, an dem den Variablen die Steuerwerte zugewiesen werden.

Mit dem Menübefehl **Variable > Trigger** können Sie den Triggerpunkt und eine Triggerbedingung einstellen

Trigger	Einstellungsmöglichkeiten
Triggerpunkt	Beginn Zyklus Ende Zyklus Übergang von RUN nach STOP.
Triggerbedingung	einmalig permanent

Triggerpunkt

Die Lage der Triggerpunkte veranschaulicht das folgende Bild.



Aus der Lage der Triggerpunkte geht hervor:

- das Steuern von Eingängen ist nur sinnvoll bei Triggerpunkt "Zyklusbeginn" (entspricht Beginn des Anwenderprogramms OB 1), da sonst das Prozeßabbild der Eingänge nach dem Steuern wieder aktualisiert und damit überschrieben wird
- das Steuern von Ausgängen ist nur sinnvoll bei Triggerpunkt "Zyklusende" (entspricht Ende des Anwenderprogramms OB 1), da sonst das Prozeßabbild der Ausgänge vom Anwenderprogramm überschrieben werden kann

Damit Ihnen der gesteuerte Wert in der Spalte "Statuswert" angezeigt wird, sollten Sie den Triggerpunkt für Beobachten auf ”Zyklusbeginn” und den Triggerpunkt für Steuern auf ”Zyklusende” einstellen!

Beim Steuern von Variablen gilt bezüglich der Triggerpunkte:

- Wenn Sie als Triggerbedingung ”Einmalig” eingestellt haben, so erhalten Sie eine Meldung, falls die ausgewählten Variablen nicht zu steuern sind.
- Bei der Triggerbedingung ”Permanent” erhalten Sie keine Rückmeldung.

Trigger sofort

Sie können die Werte ausgewählter Variablen mit dem Menübefehl **Variable > Steuerwerte aktivieren** steuern. Dieser Auftrag wird einmalig und schnellstmöglich durchgeführt, ohne Bezug zu einer bestimmten Stelle im Anwenderprogramm. Diese Funktion wird hauptsächlich im STOP-Zustand zum Steuern angewendet.

Triggerbedingung

Nachfolgende Tabelle zeigt, was die eingestellte Triggerbedingung für das Steuern von Variablen bewirkt:

	Triggerbedingung Einmalig	**Triggerbedingung Permanent**
Steuern von Variablen	*Einmaliges Aktivieren (Steuern von Variablen)* Sie können Variablen abhängig vom Triggerpunkt einmalig Werte zuweisen.	*Steuern mit definiertem Trigger* Durch das Zuweisen von festen Werten können Sie für Ihr Anwenderprogramm bestimmte Situationen simulieren und somit die programmierten Funktionen testen.

18.9 Forcen von Variablen

18.9.1 Einführung zum Forcen von Variablen

Sie können einzelne Variablen eines Anwenderprogramms mit festen Werten vorbelegen, so daß sie auch vom Anwenderprogramm, das in der CPU abläuft, nicht verändert oder überschrieben werden können. Voraussetzung ist, daß die CPU diese Möglichkeit unterstützt (wie z. B. S7-400-CPUs). Durch das feste Vorbelegen von Variablen mit Werten können Sie für Ihr Anwenderprogramm bestimmte Situationen einstellen und damit die programmierten Funktionen testen.

Fenster "Forcewerte"

Die Menüeeinträge zum Forcen sind erst anwählbar, wenn das Fenster "Forcewerte" aktiv ist.

Zur Anzeige dieses Fensters wählen Sie den Menübefehl **Variable > Forcewerte anzeigen**.

Für eine CPU sollten Sie nur ein einziges Fenster "Forcewerte öffnen". In diesem Fenster werden die Variablen mit den zugehörigen Forcewerten für den aktiven Forceauftrag angezeigt.

Beispiel eines Forcefensters

		Operand	Symbol	Anzeigeformat	Forcewert
1	F	EB 0		HEX	B#16#10
2	F	A 0.1		BOOL	true
3	F	A 1.2		BOOL	true
4					

In der **Titelzeile** steht der Name der aktuellen Online-Verbindung.

In der **Statuszeile** steht der Zeitpunkt (Datum und Uhrzeit), zu dem der Forceauftrag aus der CPU gelesen wurde.

Wenn kein Forceauftrag aktiv ist, ist das Fenster leer.

Die unterschiedliche Art der **Anzeige von Variablen** im Fenster Forcewerte hat folgende Bedeutung:

Anzeige	Bedeutung
Anzeige fett:	Variablen, die bereits fest mit einem Wert in der CPU vorbelegt sind.
Anzeige normal:	Variablen, die gerade editiert werden.
Anzeige gegraut:	Variablen einer Baugruppe, die nicht vorhanden/nicht gesteckt ist oder Variablen mit Adressierungsfehler, es wird eine Fehlermeldung eingeblendet.

Übernehmen von forcefähigen Operanden aus der Variablentabelle

Markieren Sie in der Variablentabelle die Variablen, die Sie forcen wollen.

Beim Aufblenden des Fensters "Forcewerte" werden diese Variablen übernommen, wenn die Baugruppe die Variablen forcen kann.

Übernehmen des Forceauftrags aus der CPU oder Neuerstellung eines Forceauftrags

Wenn das Fenster "Forcewerte" geöffnet und aktiv ist, wird eine weitere Meldung eingeblendet:

- Wenn Sie bestätigen, werden Änderungen im Fenster "Forcewerte" mit dem in der CPU vorhandenen Forceauftrag überschrieben. Mit dem Menübefehl **Bearbeiten > Rückgängig** können Sie den vorherigen Fensterinhalt wiederherstellen.
- Wenn Sie abbrechen, bleibt der aktuelle Inhalt des Fensters "Forcewerte" erhalten.
  Danach können Sie den Inhalt im Fenster "Forcewerte" als Variablentabelle mit dem Menübefehl **Tabelle > Speichern unter** sichern oder den Menübefehl **Variable > Forcen** anwählen: damit wird der aktuelle Inhalt des Fensters "Forcewerte" als neuer Forceauftrag in die CPU geschrieben.

Beobachten und Steuern von Variablen ist nur in der Variablentabelle möglich, nicht aber im Fenster "Forcewerte".

Speichern eines Forcewerte-Fensters

Den Inhalt eines Fensters "Forcewerte" können Sie in einer Variablentabelle sichern. Mit dem Menübefehl **Einfügen > Variablentabelle** können Sie den gesicherten Inhalt wieder in ein Forcewerte-Fenster einfügen.

Hinweise zur Symbolik im Fenster "Forcewerte"

Die Symbolik des letzten aktiven Fensters wird übernommen, außer wenn Sie "Variable beobachten und steuern" von einer anderen Applikation aufrufen, die keine Symbolik hat.

Wenn Sie keine symbolischen Namen eingeben können, ist die Spalte "Symbol" ausgeblendet. Der Menüeintrag **Extras > Symboltabelle** ist dann ebenfalls nicht anwählbar.

18.9.2 Sicherheitsvorkehrungen zum Forcen von Variablen

Personen- und Sachschaden verhüten!

Beachten Sie, daß beim Ausführen der Funktion "Forcen" eine falsche Handlung

- das Leben oder die Gesundheit von Personen gefährden oder
- Schäden an der Maschine oder an der gesamten Anlage verursachen kann.

Vorsicht

- Bevor Sie die Funktion "Forcen" starten, sollten Sie sicherstellen, daß niemand gleichzeitig auf derselben CPU diese Funktion ausführt.
- Ein Forceauftrag kann nur mit dem Menübefehl **Variable > Force löschen** gelöscht bzw. beendet werden. Das Schließen des Fensters Forcewerte oder Beenden der Applikation "Variable beobachten und steuern" löscht den Forceauftrag nicht.
- "Forcen" kann mit dem Menübefehl **Bearbeiten > Rückgängig** nicht rückgängig gemacht werden.
- Informieren Sie sich über die Unterschiede zwischen Forcen und Steuern von Variablen.
- Wenn eine CPU die Funktion Forcen nicht unterstützt, sind alle das Forcen betreffende Menübefehle im Menü Variable nicht anwählbar.

Wenn die Ausgabesperre mit dem Menübefehl **Variable > PA freischalten** aufgehoben ist, geben alle geforcten Ausgabebaugruppen ihren Forcewert aus.

18.9.3 Unterschiede zwischen Forcen und Steuern von Variablen

In der folgenden Übersicht sind die Unterschiede zwischen Forcen und Steuern zusammengefaßt:

Merkmal / Funktion	**Forcen mit S7-400**	**Forcen mit S7-300**	**Steuern**
Merker (M)	**ja**	-	**ja**
Zeiten und Zähler (T, Z)	-	-	**ja**
Datenbausteine (DB)	-	-	**ja**
Peripherie-Eingänge (PEB, PEW, PED)	**ja**	-	-
Peripherieausgänge (PAB, PAW, PAD)	**ja**	-	**ja**
Eingänge und Ausgänge (E, A)	**ja**	**ja**	**ja**
Anwenderprogramm kann die Steuer-/ Forcewerte überschreiben.	-	**ja**	**ja**
Ersetzen des Forcewertes unterbrechungsfrei wirksam	**ja**	**ja**	-
Nach Beenden der Applikation behalten die Variablen die Werte.	**ja**	**ja**	-
Nach dem Auftrennen der Verbindung zur CPU behalten die Variablen die Werte.	**ja**	**ja**	-
Adressierungsfehler erlaubt: z. B. EW1 Steuer-/Forcewert: 1 EW1 Steuer-/Forcewert: 0	-	-	der letzte wird wirksam
Trigger einstellen	immer Trigger sofort	immer Trigger sofort	Einmalig oder permanent
Funktion wirkt nur auf Variable im sichtbaren Bereich des aktiven Fensters	wirkt auf alle Forcewerte	wirkt auf alle Forcewerte	**ja**

Hinweis

- Beim "PA freischalten" werden die Forcewerte für geforcte Peripherieausgänge an den entsprechenden Ausgabebaugruppen wirksam, die Steuerwerte für permanent gesteuerte Peripherieausgänge jedoch nicht.
- Beim Forcen besitzt die Variable immer den geforcten Wert. Dieser Wert wird bei jedem Lesezugriff im Anwenderprogramm gelesen. Sämtliche Schreibzugriffe sind unwirksam.
- Beim permanenten Steuern werden Schreibzugriffe des Programms wirksam und bleiben bis zum nächsten Triggerpunkt wirksam.

19 Testen mit Programmstatus

19.1 Testen mit Programmstatus

Sie können Ihr Programm testen, indem Sie sich für jede Anweisung den Zustand des Programmstatus (VKE, Statusbit) oder den Inhalt der entsprechenden Register anzeigen lassen. Den Umfang der angezeigten Information legen Sie im Register "AWL" des Dialogfelds "Einstellungen" fest. Das Dialogfeld öffnen Sie mit dem Menübefehl **Extras > Einstellungen** im Fenster "KOP/AWL/FUP: Bausteine programmieren".

Warnung

Ein Test bei laufendem Anlagenbetrieb kann bei Funktionsstörungen oder Programmfehlern schwere Sach- und Personenschäden verursachen!

Vergewissern Sie sich, daß keine gefährlichen Zustände eintreten können, bevor Sie diese Funktion ausführen!

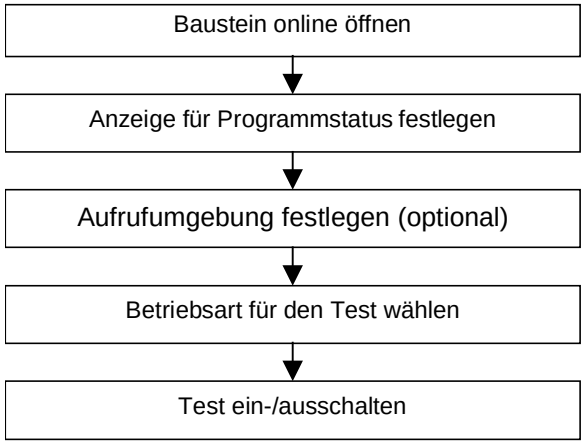
Voraussetzungen

Um den Programmstatus anzeigen lassen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sie müssen den Baustein fehlerfrei gespeichert und danach auf die CPU geladen haben.
- Die CPU ist in Betrieb, das Anwenderprogramm läuft.
- Sie haben den Baustein online geöffnet.

Prinzipielle Vorgehensweise für die Beobachtung des Programmstatus:

Es ist dringend zu empfehlen, nicht sofort das gesamte Programm aufzurufen und zu testen, sondern die Bausteine nach und nach aufzurufen und zu testen. Dabei beginnen Sie mit den unterlagerten Codebausteinen, d.h. Sie testen die Bausteine in der letzten Schachtelungstiefe der Aufrufhierarchie, z.B. indem Sie diese im OB1 aufrufen und über Beobachten und Steuern von Variablen dem Baustein die zu testende Umgebung schaffen.



Zum Setzen von Haltepunkten und zur Programmausführung in Einzelschritten muß der Betriebsmodus Testbetrieb eingestellt sein (siehe Menübefehl **Test > Betrieb**). Im Prozeßbetrieb sind diese Testfunktionen nicht möglich.

19.2 Anzeigen im Programmstatus

Die Anzeige des **Programmstatus** wird zyklisch aktualisiert. Sie beginnt mit dem markierten Netzwerk.

Voreingestellte Kennzeichnungen in KOP und FUP

- Status erfüllt: grüne durchgezogene Linien
- Status nicht erfüllt: blaue gestrichelte Linien
- Status unbekannt: schwarze durchgezogene Linien

Diese Voreinstellung von Linienart und Farbe ist änderbar über den Menübefehl **Extras > Einstellungen** / Register KOP/FUP.

Status von Elementen

- Der Status eines Kontakts ist
  - erfüllt, wenn der Operand den Wert "1" hat,
  - nicht erfüllt, wenn der Operand den Wert "0" hat,
  - unbekannt, wenn der Wert des Operanden unbekannt ist.
- Der Status von Elementen mit Freigabeausgang (ENO) entspricht dem Status eines Kontakts mit dem Wert des ENO-Ausgangs als Operanden.
- Der Status von Elementen mit Q-Ausgang entspricht dem Status eines Kontakts mit dem Wert des Operanden.
- Der Status bei CALLs ist erfüllt, wenn nach dem Aufruf das BIE-Bit gesetzt ist.
- Der Status einer Sprungoperation ist erfüllt, wenn der Sprung ausgeführt wird, d.h. wenn die Sprungbedingung erfüllt ist.
- Elemente mit Freigabeausgang (ENO) werden schwarz dargestellt, wenn der Freigabeausgang nicht beschaltet ist.

Status von Linien

- Linien sind schwarz, wenn sie nicht durchlaufen wurden oder ihr Status unbekannt ist.
- Der Status von Linien, die an der Stromschiene beginnen, ist immer erfüllt ("1").
- Der Status von Linien am Beginn von Parallelzweigen ist immer erfüllt ("1").
- Der Status der Linie nach einem Element ist erfüllt, wenn sowohl der Status der Linie vor dem Element als auch der Status des Elements erfüllt ist.
- Der Status der Linie nach NOT ist erfüllt, wenn der Status der Linie vor NOT nicht erfüllt ist (und umgekehrt).
- Der Status der Linie **nach** der Zusammenführung mehrerer Linien ist erfüllt, wenn
  - sowohl der Status von mindestens einer Linie vor der Zusammenführung erfüllt ist,
  - als auch der Status der Linie vor der Verzweigung erfüllt ist.

Status von Parametern

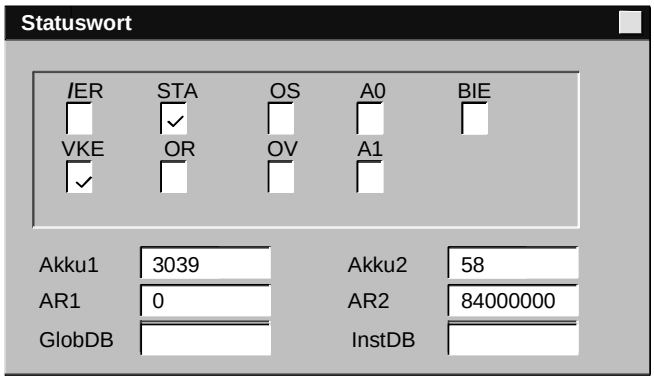
- Die Werte von Parametern **in fetter Schrift** sind aktuell.
- Die Werte von Parametern in magerer Schrift stammen aus einem vorhergehenden Zyklus; die Programmstelle wurde im aktuellen Zyklus nicht durchlaufen.

19.3 Wissenswertes zum Testen im Einzelschrittmodus / Haltepunkte

Beim Testen im Einzelschrittmodus können Sie

- Programme Anweisung für Anweisung (= Einzelschritt) bearbeiten
- Haltepunkte setzen

Die Funktion "Testen im Einzelschrittmodus" ist nicht bei allen Automatisierungssystemen realisiert (vgl. Dokumentation des entsprechenden Automatisierungssystems)



Voraussetzungen

- Der Betriebsmodus Testbetrieb muß eingestellt sein. Im Prozeßbetrieb ist ein Testen im Einzelschrittmodus nicht möglich (siehe Menübefehl **Test > Betrieb**).
- Das Testen im Einzelschrittmodus ist nur in AWL möglich. Bei Bausteine in KOP oder FUP müssen Sie über den Menübefehl **Ansicht > AWL** die Ansicht ändern.
- Der Baustein darf nicht geschützt sein.
- Der Bausteine muß online geöffnet sein.
- Der geöffnete Baustein darf nicht im Editor verändert worden sein.

Anzahl der Haltepunkte:

Die Anzahl der Haltepunkte ist variabel und ist abhängig von:

- Anzahl bereits gesetzter Haltepunkte
- Anzahl laufender Variablenstati
- Anzahl laufender Programmstati

Bitte entnehmen Sie aus Ihrer Dokumentation zu Ihrem Automatisierungssystem, ob es das Testen im Einzelschrittmodus unterstützt.

Die Menübefehle, mit denen Sie Haltepunkte setzen, aktivieren oder löschen können, finden Sie im Menü "Test". Außerdem haben Sie die Möglichkeit, diese Menübefehle über Symbole in der Haltepunktleiste auszuwählen. Um die Haltepunktleiste anzuzeigen, wählen Sie den Menübefehl **Ansicht > Haltepunktleiste**.

Zulässige Testfunktionen

- Variable beobachten/steuern
- Baugruppenzustand
- Betriebszustand

Gefahr

Achtung gefährlicher Anlagenzustand bei Betriebszustand "HALT".

19.4 Wissenswertes zum Betriebszustand HALT

Läuft das Programm auf einen Haltepunkt, so geht das Automatisierungssystem in den Betriebszustand "HALT".

Anzeige der Leuchtdioden (LEDs) im Betriebszustand Halt:

- LED RUN blinkt
- LED STOP leuchtet

Programmbearbeitung im Betriebszustand "HALT"

- Im Betriebszustand "HALT" erfolgt keine Bearbeitung des S7-Codes, d.h. alle Ablaufebenen werden nicht weiterbearbeitet.
- Alle Zeiten werden eingefroren:
  - keine Bearbeitung der Zeitzellen
  - Anhalten aller Überwachungszeiten
  - Anhalten der Grundtakte der zeitgesteuerten Ebenen
- Echtzeituhr läuft
- Aus Sicherheitsgründen wird im Betriebszustand "HALT" immer die Ausgänge abgeschaltet (vgl. Ausgabebaugruppen "output disable")

Verhalten bei Netzausfall im Betriebszustand "HALT"

- Gepufferte Automatisierungssysteme gehen bei Netzausfall während des Betriebszustandes "HALT" und anschließender Netzwiederkehr in den Betriebszustand "STOP" und bleiben dort. Die CPU führt keinen automatischen Anlauf aus. Ausgehend vom STOP können Sie das weitere Vorgehen bestimmen (z.B. Haltepunkte setzen/rücksetzen, manuellen Anlauf durchführen).
- Nicht gepufferte Automatisierungssysteme haben kein "Gedächtnis" und führen deshalb bei Netzwiederkehr einen automatischen Neustart aus, unabhängig vom vorherigem Betriebszustand.

19.5 Programmstatus von Datenbausteinen

Ab STEP 7 Version 5 ist es möglich, einen Datenbaustein in der Datensicht online zu beobachten. Die Anzeige kann sowohl von einem Online- als auch von einem Offline-Datenbaustein aktiviert werden. In beiden Fällen wird der Inhalt des Online-Datenbausteins aus dem Zielsystem dargestellt.

Der Datenbaustein darf vor dem Start des Programmstatus nicht geändert werden. Besteht ein struktureller Unterschied (Deklaration) zwischen dem Online- und dem Offline-Datenbaustein, so kann direkt per Rückfrage der Offline-Datenbaustein in das Zielsystem geladen werden.

Der Datenbaustein muß sich in der Ansicht "Datensicht" befinden, damit die Onlinewerte in der Spalte "Aktualwert" dargestellt werden können. Aktualisiert wird dabei nur der am Bildschirm sichtbare Teil des Datenbausteins. Während der Status aktiv ist kann nicht in die Deklarationssicht umgeschaltet werden.

Während der Aktualisierung ist der grüne Laufbalken in der Statuszeile des Datenbausteins sichtbar und der Betriebszustand wird eingeblendet.

Die Werte werden im Format des jeweiligen Datentyps ausgegeben, eine Änderung des Formats ist nicht möglich.

Nach der Beendigung des Programmstatus wird in der Aktualwertspalte wieder der vor dem Programmstatus gültige Inhalt dargestellt. Eine Übernahme der aktualisierten Online-Werte in den Offline-Datenbaustein ist nicht möglich.

Aktualisierung von Datentypen:

Alle elementaren Datentypen werden sowohl in einem globalen DB als auch in allen Deklarationen (in/out/inout/stat) eines Instanzdatenbausteins aktualisiert.

Einige Datentypen können nicht aktualisiert werden. Bei aktiviertem Programmstatus werden diese Felder mit nicht aktualisierten Daten in der Spalte "Aktualwert" mit grauem Hintergrund dargestellt.

- Die zusammengesetzten Datentypen DATE_AND_TIME und STRING werden nicht aktualisiert.
- In den zusammengesetzten Datentypen ARRAY, STRUCT, UDT, FB, SFB werden nur diejenigen Elemente aktualisiert, die elementare Datentypen sind.
- In der INOUT-Deklaration eines Instanzdatenbausteins wird nur der Pointer auf den zusammengesetzten Datentyp dargestellt aber nicht dessen Elemente. Der Pointer wird nicht aktualisiert.
- Parametertypen werden nicht aktualisiert.

19.6 Einstellen der Aufrufumgebung des Bausteins

Für die Aufzeichnung des Programmstatus können Sie durch das Festlegen der Aufrufumgebung genaue Bedingungen angeben. Der Programmstatus wird dann nur aufgezeichnet, wenn die angegebene Triggerbedingung erfüllt ist.

Gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Test > Aufrufumgebung**.
2. Legen Sie im angezeigten Dialogfeld die Triggerbedingungen fest und bestätigen Sie diese mit "OK".

Auswahlmöglichkeit	Bedeutung
Aufrufpfad	Hier können Sie den Aufrufpfad angeben, über den der zu testende Baustein aufgerufen werden muß, um eine Statusaufzeichnung auszulösen. Sie können die letzten drei Aufrufebenen vor Erreichen des Testbausteins eingeben.
Mit Adresse	Deaktivieren Sie diese Option, wenn die Aufrufpfadbedingung aufgehoben werden soll.
Offene Datenbausteine	Hier wird die Aufrufumgebung durch Angabe von einem oder zwei Datenbausteinen festgelegt. Die Statusaufzeichnung erfolgt, wenn der zu testende Baustein mit den jeweils angegebenen Datenbausteinen aufgerufen wurde.

Aufrufumgebung für Baustein-Instanzen festlegen

Um den Programmstatus eines Bausteins in einer bestimmten Instanz anzeigen zu lassen, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Test > Betrieb** und stellen Sie den Betriebsmodus "Testbetrieb" ein.
2. Öffnen Sie den aufrufenden Baustein und positionieren Sie den Cursor auf die gewünschte Aufrufanweisung (CALL-Zeile in AWL oder Box des Bausteins in KOP/FUP).
3. Wählen Sie über die rechte Maustaste den Menübefehl **Aufgerufenen Baustein > Beobachten mit Aufrufpfad**.

   Ergebnis: Der aufgerufene Baustein wird geöffnet, in die Triggerbedingungen des Bausteins wird der Aufruf als Kriterium eingetragen und der Status für diese Instanz des Bausteins wird aktiviert.

Bereits bestehende Triggerbedingungen für Datenbausteine bleiben unverändert bestehen.

20 Testen mit dem Simulationsprogramm S7-PLCSIM (Optionspaket)

20.1 Testen mit Simulationsprogramm (Optionspaket)

Mit dem Optionspaket Simulation können Sie Ihr Programm auf einem simulierten Automatisierungssystem, das auf Ihrem Computer bzw. Ihrem Programmiergerät (z.B. PG 740) existiert, ablaufen lassen und testen. Da die Simulation vollständig in der STEP 7-Software realisiert wird, benötigen Sie keine S7-Hardware (CPU oder Signalbaugruppen). Mit der simulierten S7-CPU können Sie Programme für S7-300 und S7-400 CPUs testen und Fehler beheben.

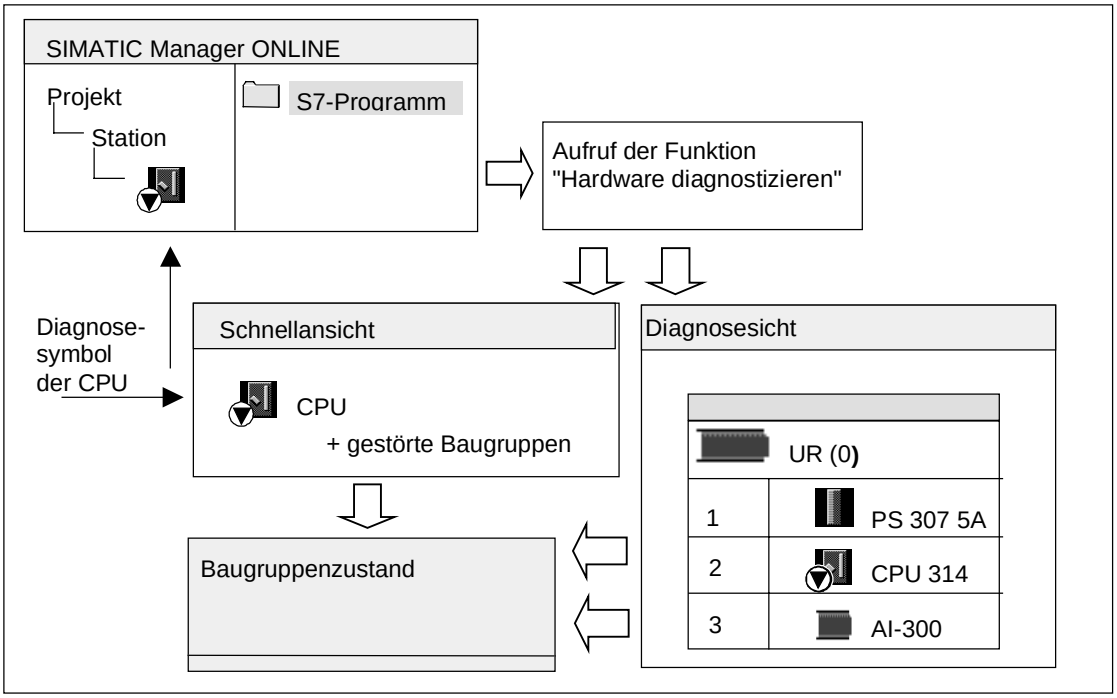
Die Applikation bietet Ihnen eine einfache Bedienoberfläche zum Überwachen und Ändern der verschiedenen Parameter, die in Ihrem Programm verwendet werden (z.B. zum Ein- und Ausschalten von Eingängen). Sie können außerdem die verschiedenen Applikationen der STEP 7-Software einsetzen, während Ihr Programm von der simulierten CPU bearbeitet wird. So können Sie beispielsweise mit der Variablentabelle Variablen überwachen und ändern.

21 Diagnose

21.1 Diagnose der Hardware und Fehlersuche

Das Vorliegen von Diagnoseinformation für eine Baugruppe können Sie anhand von Diagnosesymbolen erkennen. Diagnosesymbole zeigen den Status der zugehörigen Baugruppe und bei CPUs auch den Betriebszustand an.

Diagnosesymbole werden im Projektfenster in der Ansicht "Online" sowie nach Anfruf der Funktion "Hardware diagnostizieren" in der Schnellansicht (Voreinstellung) bzw. der Diagnosesicht angezeigt. Ausführliche Diagnoseinformationen werden im "Baugruppenzustand".angezeigt, den Sie durch Doppelklick auf ein Diagnosesymbol in der Schnellansicht bzw. Diagnosesicht aufrufen können.



Vorgehensweise zum Eingrenzen von Störungen

1. Öffnen Sie das Online-Fenster des Projekts mit dem Menübefehl **Ansicht > Online**.
2. Öffnen Sie alle Stationen, so daß die darin konfigurierten programmierbaren Baugruppen angezeigt werden.
3. Stellen Sie fest, bei welcher CPU ein Diagnosesymbol angezeigt wird, das auf einen Fehler oder eine Störung hinweist. Die Hilfeseite mit der Erklärung der Diagnosesymbole öffnen Sie mit der Taste F1.
4. Markieren Sie die Station, die Sie untersuchen möchten.
5. Wählen Sie den Menübefehl **Zielsystem > Baugruppenzustand**, um den Baugruppenzustand der CPU dieser Station anzuzeigen.
6. Wählen Sie den Menübefehl **Zielsystem > Hardware diagnostizieren**, um die "Schnellansicht" mit der CPU und den gestörten Baugruppen dieser Station anzuzeigen. Die Anzeige der Schnellansicht ist voreingestellt (Menübefehl **Extras > Einstellungen**, Register "Ansicht").
7. Markieren Sie in der Schnellansicht eine gestörte Baugruppe.
8. Klicken Sie auf die Schaltfläche "Baugruppenzustand", um die Informationen für diese Baugruppe zu erhalten.
9. Klicken Sie in der Schnellansicht die Schaltfläche "Station online öffnen", um die Diagnosesicht anzuzeigen. Die Diagnosesicht enthält alle Baugruppen der Station in der Steckplatzanordnung.
10. Doppelklicken Sie auf eine Baugruppe in der Diagnosesicht, um ihren Baugruppenzustand anzuzeigen. So erhalten Sie Informationen auch für diejenigen Baugruppen, die nicht gestört sind und deshalb in der Schnellansicht nicht angezeigt werden.

Sie müssen nicht unbedingt alle Schritte durchführen, sondern können abbrechen, sobald Sie die benötigte Diagnoseinformation erhalten haben.

21.2 Diagnosesymbole in der Online-Sicht

Diagnosesymbole werden im Online-Fenster des Projektes sowie im Fenster der Hardware-Konfiguration mit der Online-Ansicht von Konfigurationstabellen angezeigt.

Diagnosesymbole erleichtern Ihnen die Fehlersuche im Störungsfall. Sie können mit einem Blick auf ein Baugruppensymbol erkennen, ob dazu Diagnoseinformation vorliegt. Bei störungsfreiem Betrieb werden die Symbole der Baugruppentypen ohne zusätzliche Diagnosesymbole dargestellt.

Wenn zu einer Baugruppe Diagnoseinformation vorliegt, so wird zusätzlich zum Baugruppensymbol ein Diagnosesymbol angezeigt, oder das Baugruppensymbol wird kontrastvermindert dargestellt.

Diagnosesymbole für Baugruppen (Beispiel FM / CPU)

Symbol	Bedeutung
	Soll-Ist-Abweichung von der Projektierung: Die projektierte Baugruppe ist nicht vorhanden oder ein anderer Baugruppentyp ist gesteckt.
	Fehler: Baugruppe ist gestört. Mögliche Ursachen: Erkennung eines Diagnosealarms, eines Peripheriezugriffsfehlers oder einer Fehler-LED
	Es ist keine Diagnose möglich, weil keine Online-Verbindung besteht oder die CPU keine Diagnoseinformation zur Baugruppe liefert (z. B. Stromversorgung, Submodule)

Diagnosesymbole für Betriebszustände (Beispiel CPU)

Symbol	Betriebszustand
	ANLAUF
	STOP
	STOP ausgelöst durch Betriebszustand "STOP" einer anderen CPU im Multicomputing-Betrieb
	RUN
	HALT

Diagnosesymbol für das Forcen

Symbol	Betriebszustand
	Auf dieser Baugruppe wird ein Forcen von Variablen durchgeführt, d.h. Variable im Anwenderprogramm der Baugruppe sind mit festen Werten vorbelegt, die vom Programm nicht geändert werden können. Die Kennzeichnung für Forcen kann auch im Zusammenhang mit anderen Symbolen auftreten (hier mit dem Symbol für den Betriebszustand "RUN").

Aktualisieren der Anzeige von Diagnosesymbolen

Das betreffende Fenster muß aktiviert sein.

- Drücken Sie die Funktionstaste F5 oder
- wählen Sie im Fenster den Menübefehl **Ansicht > Aktualisieren**.

21.3 Hardware diagnostizieren: Schnellansicht

21.3.1 Aufruf der Schnellansicht

Die Schnellansicht bietet Ihnen einen schnellen Einstieg in ”Hardware diagnostizieren” mit reduzierter Information im Vergleich zu den ausführlicheren Anzeigen in der Diagnosesicht von HWKonfig. Die Schnellansicht wird gemäß Voreinstellung beim Aufruf der Funktion "Hardware diagnostizieren" angezeigt.

Anzeigen der Schnellansicht

Die Funktion rufen Sie aus dem SIMATIC Manager mit dem Menübefehl **Zielsystem > Hardware diagnostizieren** auf.

Der Menübefehl kann wie folgt benutzt werden:

- im Online-Fenster des Projekts, wenn eine Baugruppe oder ein S7-/M7-Programm markiert ist.
- im Fenster "Erreichbare Teilnehmer", wenn ein Teilnehmer ("MPI=...") markiert ist und dieser Eintrag zu einer CPU-Baugruppe gehört.

Aus den aufgeblendeten Konfigurationstabellen können Sie dann Baugruppen auswählen, deren Baugruppenzustand angezeigt werden soll.

21.3.2 Auskunftsfunktionen der Schnellansicht

In der Schnellansicht werden folgende Informationen angezeigt:

- Daten zur Online-Verbindung zur CPU
- Diagnosesymbol zur CPU
- Diagnosesymbole zu Baugruppen, für die die CPU eine Störung erkannt hat (z.B. Diagnosealarm, Peripheriezugriffsfehler).
- Baugruppentyp und Adresse der Baugruppe (Baugruppenträger, Steckplatz, DP Mastersystem mit Stationsnnummer)

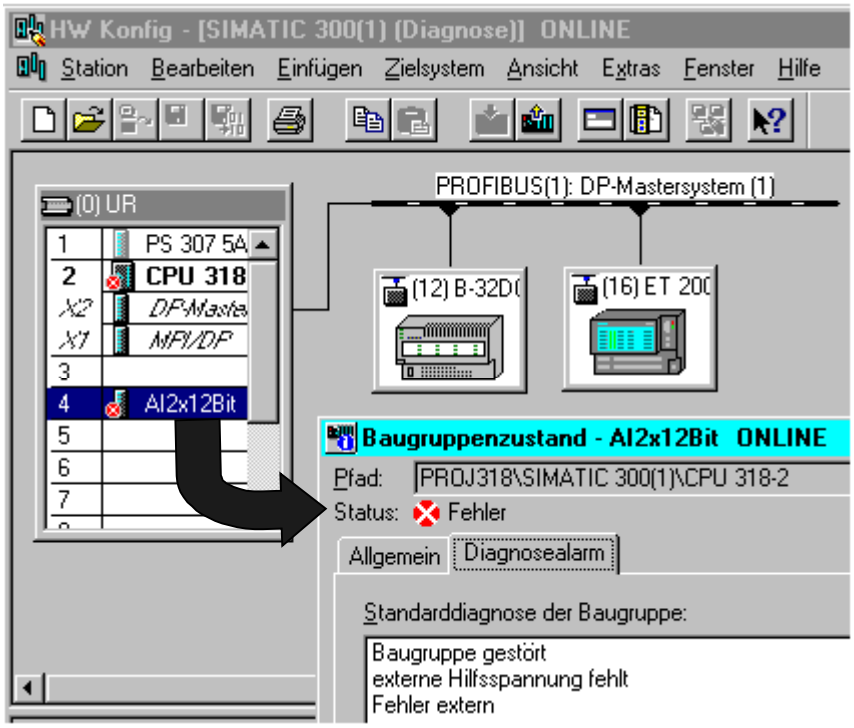
Weitere Diagnosemöglichkeiten in der Schnellansicht

- **Anzeige des Baugruppenzustands**
  Mit der Schaltfläche "Baugruppenzustand"rufen Sie dieses Dialogfeld auf. Dort erhalten Sie je nach Diagnosefähigkeit der Baugruppe detailierte Diagnoseinformationen zur markierten Baugruppe. Insbesondere können Sie sich über den Baugruppenzustand der CPU die Einträge im Diagnosepuffer anzeigen lassen.
- **Anzeige der Diagnosesicht**
  Mit der Schaltfläche "Station online öffnen"rufen Sie das Dialogfeld auf, das im Gegensatz zur Schnellansicht eine graphische Übersicht über die gesamte Station sowie Projektierinformationen enthält. Es wird auf die Baugruppe positioniert, die in der Liste "CPU / Gestörte Baugruppen" markiert ist.

21.4 Hardware diagnostizieren: Diagnosesicht

21.4.1 Aufruf der Diagnosesicht von HW Konfig

Über diesen Weg können Sie sich den Registerdialog "Baugruppenzustand" für alle Baugruppen im Baugruppenträger anzeigen lassen. Die Diagnosesicht (Konfigurationstabelle) zeigt den tatsächlichen Aufbau einer Station auf Ebene der Baugruppenträger und DP-Stationen mit Ihren Baugruppen.



Hinweis

- Ist die Konfigurationstabelle bereits offline geöffnet, so können Sie über den Menübefehl **Station > Online öffnen** ebenfalls die Online-Ansicht der Konfigurationstabellen erhalten.
- Je nach Diagnosefähigkeit der Baugruppe wird eine unterschiedliche Anzahl von Registerseiten im Registerdialog "Baugruppenzustand" angezeigt.
- Im Fenster "Erreichbare Teilnehmer" sind immer nur die Baugruppen mit eigener Teilnehmeradresse (MPI-Adresse oder PROFIBUS-Adresse) sichtbar.

Aufruf im SIMATIC Manager aus der Ansicht ONLINE eines Projekts

1. Stellen Sie in der Projektansicht des SIMATIC Managers mit dem Menübefehl **Ansicht > Online** eine Online-Verbindung zum Zielsystem her.
2. Wählen Sie eine Station aus und öffnen Sie diese durch Doppelklick.
3. Öffnen Sie anschließend das darin enthaltene Objekt "Hardware". Die Diagnosesicht wird geöffnet.

Nun können Sie eine Baugruppe markieren und deren Baugruppenzustand mit dem Menübefehl **Zielsystem > Baugruppenzustand** aufrufen.

Aufruf im SIMATIC Manager aus der Ansicht Offline eines Projekts

Gehen Sie so vor:

1. Wählen Sie aus der Projektansicht des SIMATIC Managers eine Station aus und öffnen Sie diese durch Doppelklick.
2. Öffnen Sie anschließend das darin enthaltene Objekt "Hardware". Die Konfigurationstabelle wird geöffnet.
3. Wählen Sie den Menübefehl **Station > Online öffnen**.
4. Die Diagnosesicht von HW Konfig wird geöffnet mit der Stationskonfiguration, wie sie aus den Baugruppen (z. B. CPU) ermittelt wurde. Die Zustände der Baugruppen werden durch Symbole angezeigt. Die Bedeutung der verschiedenen Symbole ersehen Sie aus der Online-Hilfe. Falls Baugruppen gestört sind oder konfigurierte Baugruppen fehlen, werden sie in einem separaten Dialogfeld aufgelistet. Von diesem Dialogfeld aus können Sie unmittelbar zu einer der angezeigten Baugruppen navigieren (Schaltfläche "Gehe zu").
5. Doppelklicken Sie auf das Symbol der Baugruppe, deren Baugruppenzustand Sie interessiert. Ein Dialogfeld mit Registern (abhängig vom Baugruppentyp) ermöglicht Ihnen eine detaillierte Analyse des Baugruppenzustands.

Aufruf im SIMATIC Manager aus dem Fenster "Erreichbare Teilnehmer"

Gehen Sie so vor:

1. Öffnen Sie im SIMATIC Manager mit dem Menübefehl **Zielsystem > Erreichbare Teilnehmer anzeigen** das Fenster "Erreichbare Teilnehmer".
2. Wählen Sie im Fenster "Erreichbare Teilnehmer" einen Teilnehmer aus.
3. Wählen Sie anschließend den Menübefehl **Zielsystem > Hardware diagnostizieren**.

Hinweis

Im Fenster "Erreichbare Teilnehmer" sind immer nur die Baugruppen mit eigener Teilnehmeradresse (MPI- oder PROFIBUS-Adresse) sichtbar.

21.4.2 Auskunftsfunktionen der Diagnosesicht

In der Diagnosesicht wird im Gegensatz zur Schnellansicht die gesamte online erreichbare Stationskonfiguration angezeigt. Diese umfaßt:

- Konfiguration der Baugruppenträger
- Diagnosesymbole zu **allen** konfigurierten Baugruppen
  Daraus ist der Status der jeweiligen Baugruppe und bei CPU-Baugruppen zusätzlich der Betriebszustand ablesbar.
- Baugruppentyp, Bestellnummer und Adreßangaben, Kommentare zur Konfigurierung

Weitere Diagnosemöglichkeiten in der Diagnosesicht

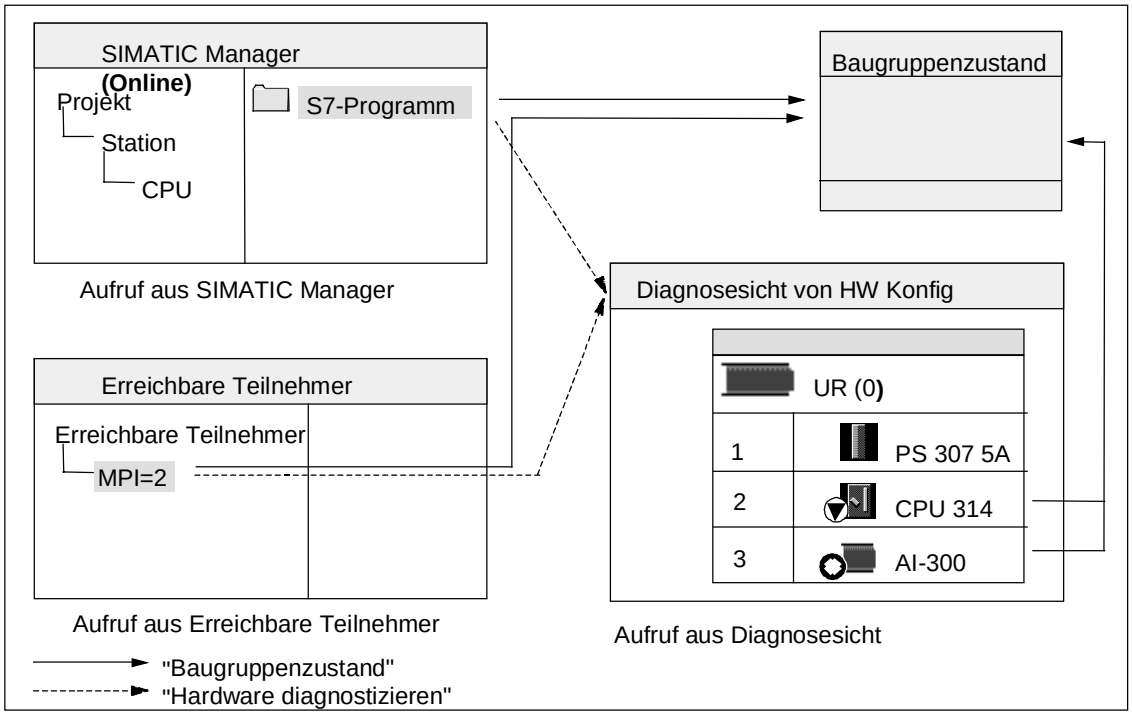
Durch Doppelklick auf eine Baugruppe können Sie sich den Baugruppenzustand zu dieser Baugruppe anzeigen lassen.

21.5 Baugruppenzustand

21.5.1.1 Aufrufmöglichkeiten des Baugruppenzustands

Den Registerdialog "Baugruppenzustand" können Sie sich von unterschiedlichen Ausgangspunkten aus anzeigen lassen. Die folgenden Vorgehensweisen sind exemplarisch und geben häufige Anwendungsfälle wieder:

- Aufruf im SIMATIC Manager aus einem Fenster mit der Projektansicht "Online" oder "Offline"
- Aufruf im SIMATIC Manager aus einem Fenster "Erreichbare Teilnehmer"
- Aufruf aus der Diagnosesicht von HW Konfig



Um den Zustand einer **Baugruppe mit eigener Teilnehmeradresse** abrufen zu können, benötigen Sie eine Online-Verbindung zum Zielsystem. Diese stellen Sie über die Online-Ansicht eines Projekts oder über das Fenster "Erreichbare Teilnehmer" her.

21.5.2 Auskunftsfunktionen des Baugruppenzustands

Die Auskunftsfunktionen stehen auf der gleichnamigen Registerseite des Dialogfelds “Baugruppenzustand” zur Verfügung. Im konkreten Anwendungsfall werden nur die Registerseiten angezeigt, die für die ausgewählte Baugruppe sinnvoll sind.

Auskunftsfunktion	Information	Einsatz
Allgemein	Identifikationsdaten der ausgewählten Baugruppe, z. B. Typ, Bestellnummer, Ausgabestand, Status, Steckplatz im Baugruppenträger.	Die Online-Auskunft aus der gesteckten Baugruppe kann mit den Daten der projektierten Baugruppe verglichen werden.
Diagnosepuffer	Übersicht über Ereignisse im Diagnosepuffer sowie detaillierte Information zum ausgewählten Ereignis	Zum Auswerten der STOP-Ursache einer CPU und zum Auswerten der vorangegangenen Ereignisse auf der ausgewählten Baugruppe. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.
Diagnosealarm	Diagnosedaten der ausgewählten Baugruppe	Zur Ermittlung der Ursache einer Baugruppenstörung
DP-Slave Diagnose	Diagnosedaten des ausgewählten DP-Slave (nach EN 50170)	Zur Ermittlung der Ursache eines Fehlers eines DP-Slaves
Speicher	Speicherausbau, Aktuelle Auslastung von Arbeitsspeicher und Ladespeicher der gewählten CPU oder M7-FM.	Vor dem Übertragen neuer oder erweiterter Bausteine auf eine CPU um zu prüfen, ob genügend Ladespeicher in dieser CPU/FM zur Verfügung steht sowie zum Komprimieren des Speicherinhalts.
Zykluszeit	Dauer des längsten, kürzesten und letzten Zyklus der gewählten CPU oder M7-FM.	Zur Kontrolle der parametrierten Mindestzykluszeit sowie der maximalen und aktuellen Zykluszeit.
Zeitsystem	Aktuelle Uhrzeit, Betriebsstunden und Informationen zur Uhrensynchronisation (Synchronisationsintervalle)	Zur Anzeige von Uhrzeit und Datum einer Baugruppe und zur Kontrolle der Uhrzeitsynchronisation
Leistungsdaten	Operandenbereiche und die verfügbaren Bausteine der angewählten (CPU-/FM-) Baugruppe	Vor und während der Erstellung eines Anwenderprogramms und zur Überprüfung, ob die CPU geeignete Voraussetzungen zum Ablauf eines Anwenderprogramms besitzt, z. B. bezüglich Größe des Prozeßabbilds.
	Anzeige aller Bausteinarten, die im Funktionsumfang der angewählten Baugruppe verfügbar sind. Liste der OBs, SFBs und SFCs, die in dieser Baugruppe eingesetzt werden können.	Zur Prüfung, welche Standard-Bausteine Ihr Anwenderprogramm enthalten bzw. aufrufen darf, um auf der angewählten CPU ablaufen zu können.

Auskunftsfunktion	Information	Einsatz
Kommunikation	Baudraten, die Verbindungs-übersicht, die Kommunikations-belastung sowie die maximale Größe der Telegramme auf dem K-Bus der angewählten Baugruppe.	Um festzustellen, wieviel und welche Verbindungen der CPU bzw. M7-FM möglich bzw. belegt sind.
Stacks	Register **Stacks**: kann nur im Betriebszustand STOP oder HALT aufgeblättert werden. Der B-Stack der angewählten Bau-gruppe wird angezeigt. Sie können außerdem U-Stack, L-Stack und Klammer-Stack auslesen und an die Fehlerstelle des unterbrechenden Bausteins springen.	Zur Feststellung der Ursache eines Übergangs in STOP und zur Korrektur eines Bausteins.

Angezeigte Zusatzinformation

Für jede Registerseite werden folgende Informationen dargestellt:

- Online-Pfad zur ausgewählten Baugruppe
- Betriebszustand der zugehörigen CPU (z. B. RUN, STOP)
- Status der ausgewählten Baugruppe (z. B. Fehler, ok)
- Betriebszustand der ausgewählten Baugruppe (z. B. RUN, STOP), soweit diese über einen eigenen Betriebszustand verfügt (z. B. CP342-5).

Der Betriebszustand der CPU selbst und der Status der ausgewählten Baugruppe können nicht angezeigt werden, wenn der Baugruppenzustand einer Nicht-CPU-Baugruppe aus dem Fenster ”Erreichbare Teilnehmer” gestartet wurde.

Gleichzeitig mehrere Baugruppen anzeigen

Sie können den Baugruppenzustand verschiedener Baugruppen abrufen und gleichzeitig darstellen. Dazu müssen Sie in den entsprechenden Baugruppenkontext wechseln, eine andere Baugruppe auswählen und den Baugruppenzustand dazu aufrufen. Es wird Ihnen dann ein weiterer Registerdialog angezeigt. Zu einer Baugruppe kann jedoch nur ein Registerdialog geöffnet werden.

Aktualisieren der Anzeigen im Baugruppenzustand

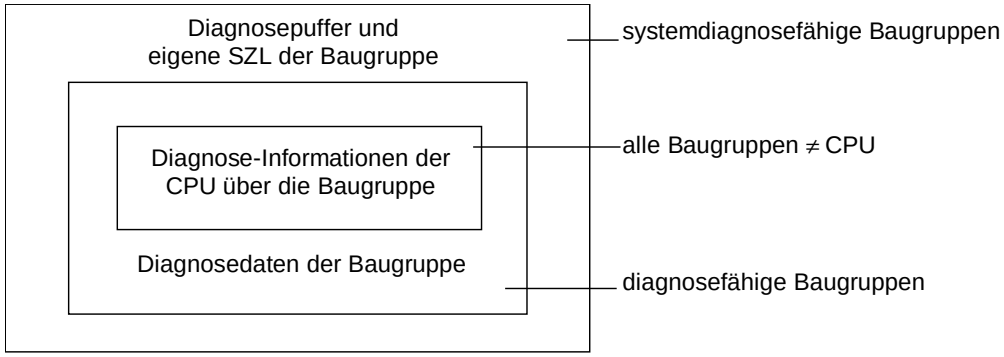
Bei jedem Wechsel auf eine andere Registerseite des Dialogfelds ”Baugruppenzustand” werden die Daten neu von der Baugruppe gelesen. Während der Anzeige einer Seite wird der Inhalt jedoch nicht automatisch aktualisiert. Durch Klicken auf die Schaltfläche ”Aktualisieren” werden die Daten erneut von der Baugruppe gelesen, ohne die Registerseite zu wechseln .

21.5.3 Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand

Der Umfang der Informationen, die ausgewertet und angezeigt werden können, ist abhängig von

- der angewählten Baugruppe und
- aus welcher Ansicht Sie den Baugruppenzustand abfragen.
  Voller Informationsumfang ist beim Aufruf aus der Online-Ansicht der Konfigurationstabellen oder aus dem Fenster des Projekts vorhanden.
  Eingeschränkter Informationsumfang ist beim Aufruf aus dem Fenster "Erreichbare Teilnehmer" möglich.

Je nach Umfang der Informationen werden die Baugruppen eingeteilt in systemdiagnosefähige, diagnosefähige oder nicht diagnosefähige Baugruppen. Diesen Zusammenhang veranschaulicht die folgende Grafik:



- Systemdiagnosefähig sind z. B. die Baugruppen FM351 und FM354.
- Diagnosefähig sind die meisten analogen SM-Baugruppen.
- Nicht diagnosefähig sind die meisten digitalen SM-Baugruppen.

Angezeigte Registerseiten

Die Tabelle zeigt, welche Registerseiten bei den einzelnen Baugruppentypen im Dialogfeld ”Baugruppenzustand” vorhanden sind.

Registerseite	**CPU oder M7-FM**	**Systemdiag-nosefähige Baugruppe**	**Diagnose-fähige Baugruppe**	**Nicht diag-nosefähige Baugruppe**	**DP-Slave**
Allgemein	ja	ja	ja	ja	ja
Diagnosepuffer	ja	ja	–	–	–
Diagnosealarm	–	ja	ja	–	ja
Speicher	ja	–	–	–	–
Zykluszeit	ja	–	–	–	–
Zeitsystem	ja	–	–	–	–
Leistungsdaten	ja	–	–	–	–
Stacks	ja	–	–	–	–
Kommunikation	ja	–	–	–	–
DP-Slave Diagnose	–	–	–	–	ja
H-Zustand [1)]	ja	–	–	–	–
[1)] nur bei CPUs in H-Systemen					

Neben der Information auf den Registerseiten wird der Betriebszustand bei Baugruppen mit Betriebszustand angezeigt. Beim Aufruf aus den Konfigurationstabellen online wird der Status angegeben, den diese Baugruppe aus Sicht der CPU besitzt (z. B. OK, Fehler, Baugruppe nicht vorhanden).

21.6 Diagnose im Betriebszustand STOP

21.6.1 Prinzipielles Vorgehen zum Ermitteln einer STOP-Ursache

Um zu ermitteln, warum die CPU in den Betriebszustand "STOP" gewechselt hat, gehen Sie wie folgt vor:

1. Markieren Sie die CPU, die in den Betriebszustand STOP gegangen ist.
2. Wählen Sie den Menübefehl **Zielsystem > Baugruppenzustand**.
3. Wählen Sie die Registerseite “Diagnosepuffer”.
4. Anhand der letzten Einträge können Sie die STOP-Ursache ermitteln.

Bei Programmierfehler:

1. Der Eintrag “STOP durch nicht geladenen Programmierfehler-OB” bedeutet beispielsweise, daß die CPU einen Programmierfehler erkannt und danach versucht hat, den (nicht vorhandenen) OB zur Behandlung des Programmierfehlers zu starten. Auf den eigentlichen Programmierfehler weist der vorhergehende Eintrag hin.
2. Markieren Sie die Meldung zum Programmierfehler.
3. Betätigen Sie die Schaltfläche "Baustein öffnen".
4. Wählen Sie die Registerseite "Stacks".

21.6.2 Stack-Inhalte im Betriebszustand STOP

Durch Auswertung des Diagnosepuffers und der Stack-Inhalte können Sie die Ursache für die Störung im Ablauf eines Anwenderprogrammes ermitteln.

Wenn die CPU z.B. durch einen Programmierfehler oder durch den Stoppbefehl in den Betriebszustand "STOP" gewechselt hat, wird auf der Registerseite “Stacks” des Baugruppenzustands der Baustein-Stack angezeigt. Weitere Stack-Inhalte können Sie über die Schaltflächen ”U-Stack”, ”L-Stack” und ”Klammer-Stack” anzeigen. Die Stack-Inhalte geben Ihnen Informationen darüber, welcher Befehl in welchem Baustein zu dem STOP bei der CPU geführt hat.

B-Stack-Inhalt

Im B-Stack werden alle Bausteine aufgeführt, die bis zum Übergang in den STOP-Zustand aufgerufen wurden und noch nicht zu Ende bearbeitet worden sind.

U-Stack-Inhalt

Durch Klicken auf die Schaltfläche ”U-Stack” werden Ihnen Daten zur Unterbrechungsstelle angezeigt. Der Unterbrechungsstack (U-Stack) enthält Daten bzw. Zustände, die zum Zeitpunkt der Unterbrechung gültig waren, z. B.

- Akku-Inhalte und Registerinhalte
- Offene DBs und deren Größe
- Inhalt des Statusworts
- Prioritätsklasse
- Unterbrochenen Baustein
- Baustein, in dem der Programmablauf nach der Unterbrechung fortgesetzt würde.

L-Stack-Inhalt

Für jeden im B-Stack aufgelisteten Baustein können Sie sich durch Selektieren des Bausteins und Klicken auf die Schaltfläche ”L-Stack” die zugehörigen Lokaldaten anzeigen lassen.

Der Lokaldaten-Stack (L-Stack) enthält die Lokaldatenwerte der Bausteine, mit denen das Anwenderprogramm bis zum Zeitpunkt der Unterbrechung gearbeitet hat.

Für die Interpretation und Auswertung der angezeigten Lokaldaten sind sehr gute Systemkenntnisse erforderlich. Der vordere Teil der angezeigten Daten entspricht den temporären Variablen des Bausteins.

Klammerstack-Inhalt

Wenn Sie die Schaltfläche ”Klammer-Stack” betätigen, wird der Inhalt des Klammer-Stack an der Unterbrechungsstelle dargestellt.

Der Klammer-Stack ist ein Speicherbereich, den die Verknüpfungsoperationen **U(**, **UN(**, **O(**, **ON(**, **X(** und **XN(** verwenden.

Die Schaltfläche ist nur aktiv, wenn zum Zeitpunkt der Unterbrechung noch offene Klammerausdrücke existierten.

21.7 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern

21.7.1 Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern

Die Registerseite “Zykluszeit” des Baugruppenzustands gibt Ihnen Auskunft über die Zykluszeiten des Anwenderprogramms.

Wenn die Dauer des längsten Zyklus dicht an der Überwachungszeit liegt, besteht die Gefahr, daß Schwankungen in der Zykluszeit zu einem Zeitfehler führen. Dies können Sie vermeiden, indem Sie die maximale Zykluszeit des Anwenderprogramms vergrößern.

Wenn die Zyklusdauer die parametrierte Mindestzykluszeit unterschreitet, wird der Zyklus automatisch von der CPU/FM auf die parametrierte Mindestzykluszeit verlängert. Im Falle einer CPU wird während dieser Verlängerung der Hintergrund-OB (OB 90) bearbeitet, falls er geladen ist.

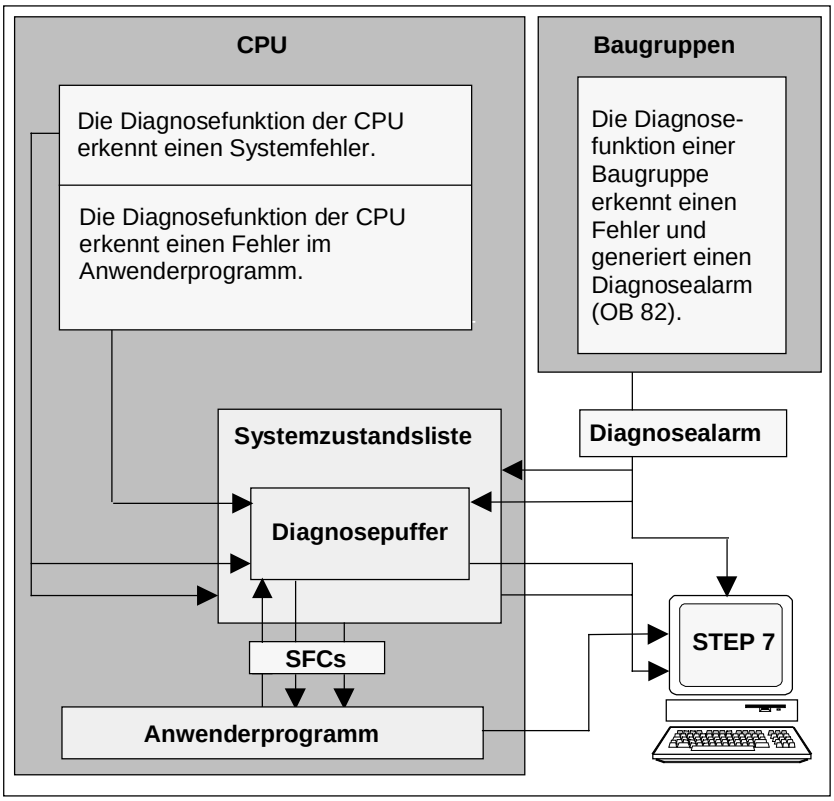
Einstellen der Zykluszeit

Die maximale und minimale Zykluszeit können Sie bei der Konfigurierung der Hardware festlegen. Dazu markieren Sie in der Offline-Ansicht der Konfigurationstabelle die CPU/FM aus und wählen über die rechte Maustaste den Menübefehl **Objekteigenschaften**, um ihre Eigenschaften zu definieren. Auf der Registerseite ”Zyklus/Taktmerker” können Sie die Werte einstellen.

21.8 Übermittlung von Diagnoseinformationen

21.8.1 Übermittlung von Diagnoseinformationen

Das folgende Bild zeigt die Übermittlung von Diagnoseinformationen bei SIMATIC S7.



Auslesen der Diagnoseinformation

Sie können die Diagnoseeinträge im Anwenderprogramm mit der SFC 51 RDSYSST auslesen oder sich die Diagnosemeldungen mit STEP 7 im Klartext anzeigen lassen.

Sie geben Auskunft darüber:

- wo und wann der Fehler aufgetreten ist
- zu welcher Art von Diagnoseereignissen der Eintrag gehört (anwenderdefiniertes Diagnoseereignis, synchroner/asynchroner Fehler, Betriebszustandswechsel).

Erzeugung von Leittechniksammelmeldungen

Die CPU trägt die Ereignisse der Standarddiagnose und der erweiterten Diagnose in den Diagnosepuffer ein. Darüber hinaus erzeugt sie bei den Standard-Diagnoseereignissen eine Leittechniksammelmeldung, falls die folgenden Voraussetzungen erfüllt sind:

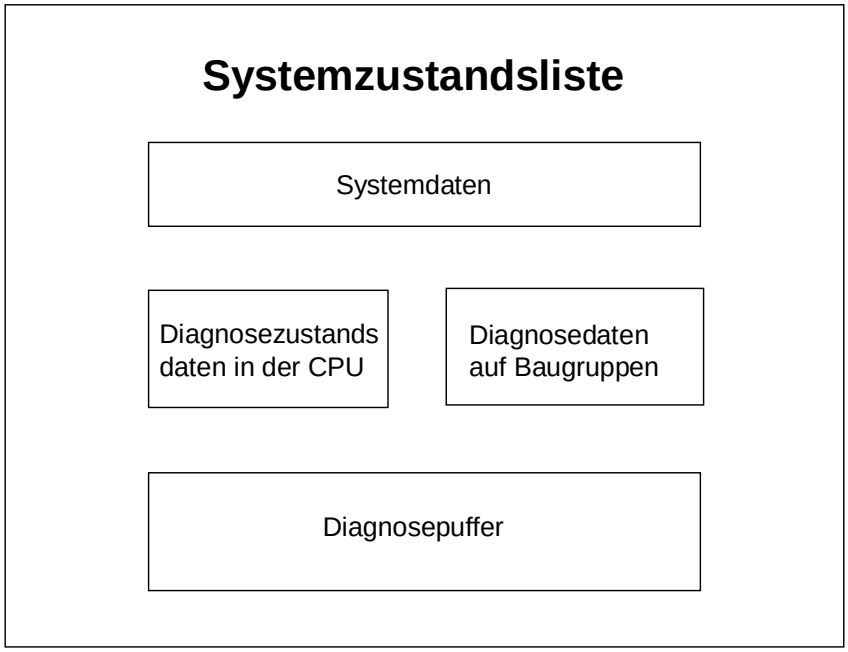
- Sie haben mit STEP 7 vorgegeben, daß Leittechniksammelmeldungen erzeugt werden sollen.
- Mindestens ein Anzeigegerät hat sich bei der CPU für Leittechniksammelmeldungen angemeldet.
- Eine Leittechniksammelmeldung wird erst dann gebildet, wenn noch keine Leittechniksammelmeldung der zugehörigen Klasse (es gibt sieben Klassen) vorliegt.
- Pro Klasse kann eine Leittechniksammelmeldung gebildet werden.

21.8.2 Systemzustandsliste SZL

Die Systemzustandsliste SZL beschreibt den aktuellen Zustand des Automatisierungssystems: sie gibt einen Überblick über den Ausbau, die aktuelle Parametrierung, die aktuellen Zustände und Abläufe in der CPU und den zugeordneten Baugruppen.

Die Daten der SZL können nur gelesen, nicht geändert werden. Sie ist eine virtuelle Liste, die nur auf Anforderung zusammengestellt wird.

Die Informationen, die Sie über die SZL ausgeben können, lassen sich in vier Bereiche aufteilen:



Systemzustandsliste lesen

Es gibt zwei Möglichkeiten, die Informationen der Systemzustandsliste SZL zu lesen:

- implizit über STEP 7-Menübefehle vom Programmiergerät aus (z. B. Speicherausbau, statische CPU-Daten, Diagnosepuffer, Zustandsanzeigen).
- explizit über die Systemfunktion SFC 51 RDSYSST vom Anwenderprogramm aus mittels Angabe der gewünschten Teillistennummer (siehe Baustein-Hilfen).

Systemdaten der SZL

Systemdaten sind feste oder parametrierte Kenndaten einer CPU. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppen-Identifikation	Bestellnummer, Typkennung und Ausgabestand der Baugruppe
CPU-Merkmale	Zeitsystem, Systemverhalten (z. B. Multicomputing) und Sprachbeschreibung der CPU
Speicherbereiche	Speicherausbau der Baugruppe (Größe des Arbeitsspeichers)
Systembereiche	Systemspeicher der Baugruppe (z. B. Anzahl der Merker, Zeiten, Zähler, Speichertyp)
Bausteintypen	Welche Bausteintypen (OB, DB, SDB, FC, FB) in Baugruppe vorhanden, wieviel Bausteine eines Typs maximal vorhanden und maximale Größe eines Bausteintyps
Alarm-Fehlerzuordnung	Zuordnung Alarme/Fehler zu OBs
Alarmstatus	Aktuelle Alarmbearbeitung/-generierung
Status der Prioritätsklassen	Welcher OB in Bearbeitung, welche Prioritätsklasse durch Parametrierung gesperrt
Betriebszustand und Betriebszustandsübergang	Welche Betriebszustände möglich, letzter Übergang, aktueller Betriebszustand

Diagnosezustandsdaten in der CPU

Diagnosezustandsdaten beschreiben den aktuellen Zustand der Komponenten, die durch die Systemdiagnose überwacht werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können:

Bereich	Information
Kommunikations-zustandsdaten	Aktuell im System eingestellte Kommunikationsfunktionen
Diagnoseteilnehmer	An der CPU angemeldete diagnosefähige Baugruppen
Startinformationsliste des OB	Startinformationen zu den OBs der CPU
Startereignisliste	Startereignisse und Prioritätsklassen der OBs
Baugruppenzustands-information	Zustandsinformationen aller gesteckten, gestörten, prozeßalarmerzeugenden, zugeordneten Baugruppen

Diagnosedaten auf Baugruppen

Außer der CPU gibt es noch andere diagnosefähige Baugruppen (SMs, CPs, FMs), deren Diagnosedaten in die Systemzustandsliste eingetragen werden. Die folgende Tabelle zeigt, zu welchen Themenbereichen Informationen (Teillisten der SZL) ausgegeben werden können.

Bereich	Information
Baugruppendiagnoseinfo	Baugruppenanfangsadresse, interne/externe Fehler, Kanalfehler, Parameterfehler (4 Byte)
Baugruppendiagnosedaten	Alle Diagnosedaten einer bestimmten Baugruppe

21.8.3 Eigene Diagnosemeldungen senden

Sie können die standardmäßige Systemdiagnose von SIMATIC S7 außerdem erweitern, indem Sie über die Systemfunktion SFC 52 WR_USMSG

- anwenderdefinierte Diagnoseereignisse (z. B. Informationen zum Ablauf des Anwenderprogramms) in den Diagnosepuffer eintragen
- anwenderdefinierte Diagnosemeldungen an angemeldete Teilnehmer (Beobachtungsgeräte wie PG, OP, TD) senden.

Anwenderdefinierte Diagnoseereignisse

Die Diagnoseereignisse sind in Ereignisklassen von 1 bis F eingeteilt. Die anwenderdefinierten Diagnoseereignisse gehören zu den Ereignisklassen 8 bis B. Sie lassen sich in zwei Gruppen unterteilen:

- Die Ereignisklassen 8 und 9 umfassen Ereignisse mit festgelegter Nummer und vorgefertigtem Text, den Sie anhand der Nummer abrufen können.
- Die Ereignisklassen A und B umfassen Ereignisse mit frei wählbarer Nummer (A000 bis A0FF, B000 bis B0FF) und frei wählbarem Text.

Diagnosemeldungen an Teilnehmer senden

Zusätzlich zum Eintrag eines anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer können Sie mit der SFC 52 WR_USMSG Ihre anwenderdefinierten Diagnosemeldungen auch noch an angemeldete Anzeigegeräte senden. Bei Aufruf der SFC 52 mit SEND = 1 wird diese Diagnosemeldung in den Sendepuffer geschrieben und automatisch an den oder die an der CPU angemeldeten Teilnehmer geschickt.

Ist ein Senden nicht möglich (z. B. weil kein Teilnehmer angemeldet wurde oder der Sendepuffer voll ist), erfolgt dennoch der Eintrag des anwenderdefinierten Diagnoseereignisses in den Diagnosepuffer.

Meldung mit Quittieranzeige erzeugen

Falls Sie ein anwenderdefiniertes Diagnoseereignis quittieren und diese Quittierung programmtechnisch erfassen wollen, gehen Sie wie folgt vor:

- Beim kommenden Ereignis beschreiben Sie eine Variable vom Typ BOOL mit 1, beim gehenden Ereignis beschreiben Sie diese mit 0.
- Diese Variable überwachen Sie mit Hilfe des SFB 33 ALARM.

21.8.4 Diagnosefunktionen

Die Systemdiagnose ist die Erkennung, Auswertung und Meldung von Fehlern, die innerhalb eines Automatisierungssystems auftreten. Dazu gibt es in jeder CPU und jeder systemdiagnosefähigen Baugruppe (z. B. FM354) einen Diagnosepuffer, in den nähere Informationen zu allen Diagnoseereignissen in der Reihenfolge ihres Auftretens eingetragen werden.

Diagnoseereignisse

Als Diagnoseereignisse werden Ihnen folgende Einträge angezeigt, z. B.:

- interne und externe Fehler auf einer Baugruppe
- Systemfehler in der CPU
- Betriebszustandsübergänge (z. B. von RUN nach STOP)
- Fehler im Anwenderprogramm
- Ziehen/Stecken von Baugruppen
- mit der Systemfunktion SFC 52 eingetragene Anwendermeldungen

Beim Urlöschen von CPUs bleibt der Inhalt des Diagnosepuffers erhalten. Fehler im System können durch den Diagnosepuffer auch nach längerer Zeit noch ausgewertet werden, um die Ursache für einen STOP festzustellen oder um das Auftreten einzelner Diagnoseereignisse zurückzuverfolgen und zuordnen zu können.

Aufzeichnung der Diagnosedaten

Die Erfassung von Diagnosedaten durch die Systemdiagnose muß nicht programmiert werden, sie ist standardmäßig vorhanden und läuft automatisch ab. SIMATIC S7 bietet verschiedene Diagnosefunktionen. Einige sind in die CPU integriert, andere werden von den Baugruppen (SMs, CPs und FMs) zur Verfügung gestellt.

Anzeige von Fehlern

Baugruppeninterne und -externe Fehler werden über LED-Anzeigen auf der Frontplatten der jeweiligen Baugruppe angezeigt. Die LED-Anzeigen und ihre Auswertung ist beschrieben in den Handbüchern zur S7-Hardware. Bei S7-300 werden interne und externe Fehler zu einem Sammelfehler zusammengefaßt.

Die CPU erkennt Systemfehler sowie Fehler im Anwenderprogramm und trägt Diagnoseereignisse in die Systemzustandsliste und den Diagnosepuffer ein. Diese Diagnosemeldungen können am PG ausgelesen werden.

Diagnosefähige Signal- und Funktionsbaugruppen erkennen interne und externe Baugruppenfehler und erzeugen einen Diagnosealarm, auf den Sie mit Hilfe eines Alarm-OBs reagieren können.

21.9 Maßnahmen im Programm zur Störungsbehandlung

21.9.1 Maßnahmen im Programm zur Störungsbehandlung

Die CPU ruft bei Erkennung von Fehlern im Programmablauf (Synchronfehler) und von Fehlern im Automatisierungssytem (Asynchronfehler) den zu dem jeweiligen Fehler gehörenden Fehler-OB auf:

Aufgetretener Fehler	**Fehler-OB**
Peripherie-Redundanzfehler	OB 70
CPU-Redundanzfehler	OB 72
Kommunikations-Redundanzfehler	OB 73
Zeitfehler	OB 80
Stromversorgungsfehler	OB 81
Diagnosealarm	OB 82
Ziehen/Stecken-Alarm	OB 83
CPU-Hardwarefehler	OB 84
Programmablauffehler	OB 85
Baugruppenträgerausfall oder Ausfall einer Station bei dezentraler Peripherie	OB 86
Kommunikationsfehler	OB 87
Programmierfehler	OB 121
Peripheriezugriffsfehler	OB 122

Ist der entsprechende OB nicht vorhanden, geht die CPU in den Betriebszustand "STOP". Andernfalls besteht die Möglichkeit, in dem OB Anweisungen zu hinterlegen, wie auf diese Fehlersituation reagiert werden soll. Dadurch können evtl. die Auswirkungen des Fehlers gemindert oder aufgehoben werden.

Prinzipielle Vorgehensweise

Anlegen und Öffnen des OB

1. Rufen Sie den Baugruppenzustand Ihrer CPU auf.
2. Wählen Sie das Register "Leistungsdaten".
3. Entnehmen Sie der angezeigten Liste, ob der zu programmierende OB für diese CPU zulässig ist.
4. Fügen Sie den OB in den Ordner "Bausteine" Ihres Programms ein und öffnen Sie diesen.
5. Geben Sie das Programm zur Behandlung des Fehlers ein.
6. Laden Sie den OB auf das Zielsystem.

Programmieren der Maßnahmen zur Fehlerbehandlung

1. Lokaldaten des OB auswerten, um genauere Fehlerursache zu ermitteln. Die Variablen OB8x_FLT_ID bzw. OB12x_SW_FLT der Lokaldaten enthalten den Fehlercode. Die Bedeutung ist im Referenzhandbuch System- und Standardfunktionen beschrieben.
2. In den Programmteil verzweigen, mit dem auf diesen Fehler reagiert wird.

Ein Beispiel zur Behandlung von Diagnosealarmen enthält die Referenzhilfe zu System- und Standardfunktionen unter dem Titel “Beispiel zur Baugruppendiagnose mit der SFC 51 (RDSYSST)”.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.2 Auswerten des Ausgangsparameter RET_VAL

Eine Systemfunktion zeigt über den Ausgangsparameter RET_VAL (Rückgabewert) an, ob die CPU die Funktion der SFC erfolgreich ausführen konnte oder nicht.

Fehlerinformationen im Rückgabewert

Der Rückgabewert ist vom Datentyp Ganzzahl (INT). Das Vorzeichen einer Ganzzahl gibt an, ob es sich um eine positive oder negative Ganzzahl handelt. Die Relation des Rückgabewerts zu dem Wert ”0” zeigt an, ob während der Bearbeitung der Funktion ein Fehler aufgetreten ist (siehe auch Tabelle 11-5):

- Tritt während der Bearbeitung der Funktion ein Fehler auf, ist der Rückgabewert kleiner als Null. Das Vorzeichenbit der Ganzzahl ist ”1”.
- Wird die Funktion fehlerfrei bearbeitet, ist der Rückgabewert größer gleich Null. Das Vorzeichenbit der Ganzzahl ist ”0”.

Bearbeitung der SFC durch die CPU	Rückgabewert	Vorzeichen der Ganzzahl
fehlerhaft	kleiner als ”0”	negativ (Vorzeichenbit ist ”1”)
fehlerfrei	größer oder gleich ”0”	positiv (Vorzeichenbit ist ”0”)

Reagieren auf Fehlerinformationen

Tritt während der Bearbeitung einer SFC ein Fehler auf, dann stellt die SFC über den Rückgabewert RET_VAL einen Fehlercode zur Verfügung.

Man unterscheidet dabei zwischen:

- einem allgemeinen Fehlercode, den alle SFCs ausgeben können und
- einem spezifischen Fehlercode, den eine SFC abhängig von seinen spezifischen Funktionen ausgeben kann.

Funktionswertübergabe

Einige SFCs benutzen den Ausgangsparameter RET_VAL auch zur Übergabe des Funktionswertes, z. B. übergibt die SFC 64 TIME_TCK mit RET_VAL die gelesene Systemzeit.

Ausführliche Informationen zum Parameter RET_VAL finden Sie in der Hilfe zu SFBs/SFCs.

21.9.3 Fehler-OBs als Reaktion auf die Erkennung eines Fehlers

Feststellbare Fehler

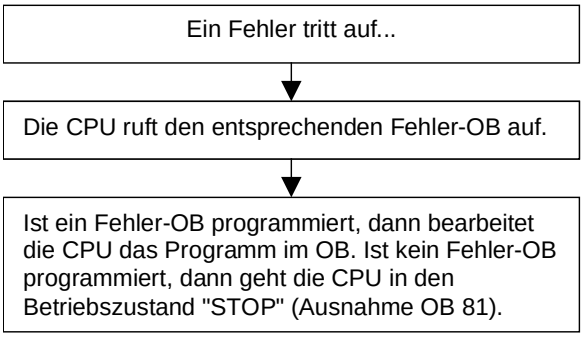
Das Systemprogramm kann folgende Fehler feststellen:

- fehlerhaftes Arbeiten der CPU
- Fehler in der Systemprogrammbearbeitung
- Fehler im Anwenderprogramm
- Fehler in der Peripherie

Je nach Fehlerart wird die CPU in den STOP-Zustand gesetzt oder ein Fehler-OB aufgerufen.

Programmieren von Reaktionen

Sie können Programme entwerfen, um auf die verschiedenen Arten von Fehlern zu reagieren und das weitere Verhalten der CPU zu bestimmen. Das Programm für einen bestimmten Fehler kann dann in einem Fehler-OB gespeichert werden. Wenn der Fehler-OB aufgerufen wird, wird das Programm bearbeitet.


Ein Fehler tritt auf...
Die CPU ruft den entsprechenden Fehler-OB auf.
Ist ein Fehler-OB programmiert, dann bearbeitet die CPU das Programm im OB. Ist kein Fehler-OB programmiert, dann geht die CPU in den Betriebszustand "STOP" (Ausnahme OB 81).

Fehler-OBs

Es wird unterschieden zwischen synchronen und asynchronen Fehlern:

- Synchrone Fehler können einem MC7-Befehl zugeordnet werden (z. B. Ladebefehl auf gezogene Signalbaugruppe).
- Asynchrone Fehler können einer Prioritätsklasse oder dem gesamten Automatisierungssystem zugeordnet werden (z. B. Zykluszeitüberlauf).

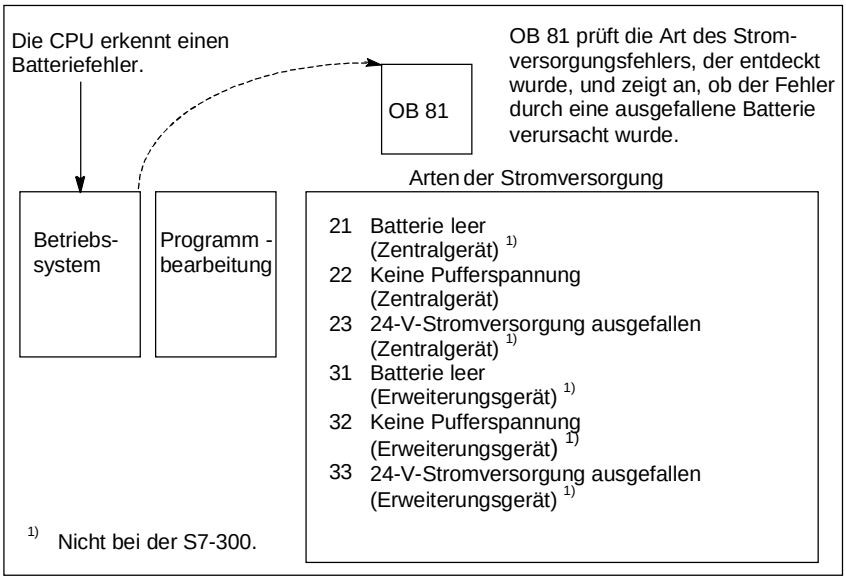
Nachfolgende Tabelle zeigt welche Fehlerarten prinzipiell auftreten können. Ob Ihre CPU die angegebenen OBs anbietet, entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" oder dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Aufbauen".

Fehlertyp	Fehlerart	OB	Priorität
Redundanz	Peripherie-Redundanzfehler (nur in H-CPUs)	OB 70	25
	CPU-Redundanzfehler (nur in H-CPUs)	OB 72	28
	Kommunikations-Redundanzfehler	OB 73	25
Asynchron	Zeitfehler	OB 80	26
	Stromversorgungsfehler	OB 81	(bzw. 28, wenn der Fehler-OB im
	Diagnosealarm	OB 82	Anlaufprogramm vorkommt)
	Ziehen/Stecken-Alarm	OB 83	
	CPU-Hardwarefehler	OB 84	
	Programmablauffehler	OB 85	
	Baugruppenträgerausfall	OB 86	
	Kommunikationsfehler	OB 87	
Synchron	Programmierfehler	OB 121	Priorität des fehlerverursachenden OBs
	Zugriffsfehler	OB 122	

Beispiel zur Verwendung des Fehler-OB 81

Sie können mit den Lokaldaten (Startinformation) des Fehler-OB die Art des aufgetretenen Fehlers auswerten.

Wenn die CPU beispielsweise einen Batteriefehler erkennt, dann ruft das Betriebssystem OB 81 auf (siehe Bild).


Die CPU erkennt einen Batteriefehler.
OB 81
OB 81 prüft die Art des Stromversorgungsfehlers, der entdeckt wurde, und zeigt an, ob der Fehler durch eine ausgefallene Batterie verursacht wurde.
Arten der Stromversorgung
Betriebs-system
Programm - bearbeitung
21 Batterie leer (Zentralgerät) 1)
22 Keine Pufferspannung (Zentralgerät)
23 24-V-Stromversorgung ausgefallen (Zentralgerät) 1)
31 Batterie leer (Erweiterungsgerät) 1)
32 Keine Pufferspannung (Erweiterungsgerät) 1)
33 24-V-Stromversorgung ausgefallen (Erweiterungsgerät) 1)
1) Nicht bei der S7-300.

Sie können ein Programm schreiben, das den Ereigniscode, der den Aufruf von OB 81 ausgelöst hat, auswertet. Sie können auch ein Programm schreiben, das eine Reaktion hervorruft, wie beispielsweise das Einschalten eines Ausgangs, der an eine Lampe der Operator-Station angeschlossen ist.

Lokaldaten des Fehler-OB 81

Tabelle 11-7 beschreibt die temporären (TEMP) Variablen, die in der Variablendeklarationstabelle des OB 81 deklariert sind.

In der Symboltabelle muß auch das Symbol *Batteriefehler* (BOOL) als Ausgang gekennzeichnet werden (z. B. A 4.0), so daß andere Teile des Programms auf diese Daten zugreifen können.

Dekl.	Name	Typ	Beschreibung
TEMP	OB81_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 39xx
TEMP	OB81_FLT_ID	BYTE	Fehlercode: b#16#21 = Mindestens eine Pufferbatterie des Zentralgeräts leer [1] b#16#22 = Pufferspannung im Zentralgerät fehlt b#16#23 = Ausfall der 24-V-Stromversorgung im Zentralgerät [1] b#16#31 = Mindestens eine Pufferbatterie eines Erweiterungsgeräts leer [1)] b#16#32 = Pufferspannung in einem Erweiterungsgerät fehlt [1)] b#16#33 = Ausfall der 24-V-Stromversorgung eines Erweiterungsgeräts [1)]
TEMP	OB81_PRIORITY	BYTE	Prioritätsklasse = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Reserviert
TEMP	OB81_RESERVED_2	BYTE	Reserviert
TEMP	OB81_MDL_ADDR	INT	Reserviert
TEMP	OB81_RESERVED_3	BYTE	Nur relevant für die Fehlercodes B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde

[1)] = Nicht bei der S7-300.

Beispielprogramm für den Fehler-OB 81

Anhand eines AWL-Beispielprogramms wird gezeigt, wie Sie den Fehlercode in OB 81 lesen können.

Das Programm ist wie folgt aufgebaut:

- Der Fehlercode in OB 81 (OB81_FLT_ID) wird gelesen und verglichen mit dem Wert des Ereignisses “Batterie leer” (B#16#3921).
- Entspricht der Fehlercode dem Code für “Batterie leer”, dann springt das Programm zur Marke BFeh und schaltet den Ausgang *Batteriefehler* ein.
- Entspricht der Fehlercode nicht dem Code für “Batterie leer”, dann vergleicht das Programm den Code mit dem Code für “Batterieausfall”.
- Entspricht der Fehlercode dem Code für “Batterieausfall”, dann springt das Programm zur Marke BFeh und schaltet den Ausgang *Batteriefehler* ein. Andernfalls wird der Baustein beendet.

```
          AWL                    Beschreibung
    L       B#16#21          // Vergleiche Ereigniscode "Batterie leer"
                             //(B#16#21) mit
    L       #OB81_FLT_ID     // dem Fehlercode für OB 81.
    ==I                      // Wenn gleich (Batterie ist leer),
                             //dann springe zu BF.
    SPB BF
    L       B#16#22          // Vergleiche Ereigniscode "Batterieausfall"
                             //(b#16#22) mit
    ==I                      // dem Fehlercode für OB 81.
    SPB BF                   // Wenn gleich, dann springe zu BF.
    BEA                      // keine Meldung über Batteriefehler

BF:  L       B#16#39         // Vergleiche Kennung für kommendes Ereignis
    L        #OB81_EV_CLASS  // mit dem Fehlercode für OB 81.
    ==I                      // wenn Batterieausfall oder leere Batterie
                             //entdeckt wird,
    S       Batteriefehler   // setze Batteriefehler
                             // (Variable aus Symboltabelle)
    L       B#16#38          // Vergleiche Kennung für gehendes Ereignis mit
    ==I                      // dem Fehlercode für OB 81.
    R      Batteriefehler    // rücksetze Batteriefehler, wenn behoben.
```

Ausführliche Informationen zu OBs, SFBs und SFCs und Erläuterungen zu den Ereignis-IDs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.4 Einfügen von Ersatzwerten bei Fehlererkennung

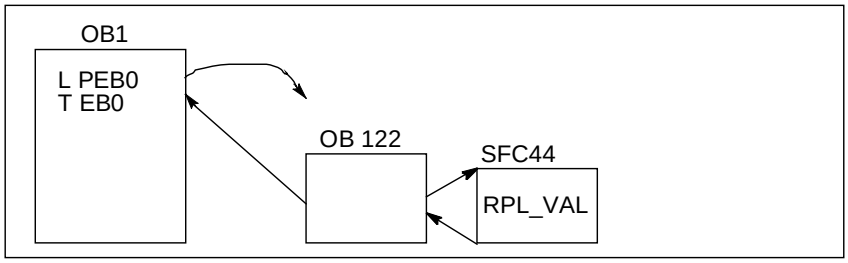
Für bestimmte Fehlerarten (z. B. Drahtbruch bei Eingangssignal) können Sie Ersatzwerte vorgeben für Werte, die aufgrund des Fehlers nicht verfügbar sind. Es gibt zwei Möglichkeiten Ersatzwerte vorzugeben:

- Sie können mit STEP 7 Ersatzwerte für parametrierbare Ausgabebaugruppen parametrieren. Nicht parametrierbare Ausgabebaugruppen haben den voreingestellten Ersatzwert 0.
- Sie können mit Hilfe der SFC 44 RPL_VAL in Fehler-OBs Ersatzwerte programmieren (nur für Eingabebaugruppen).

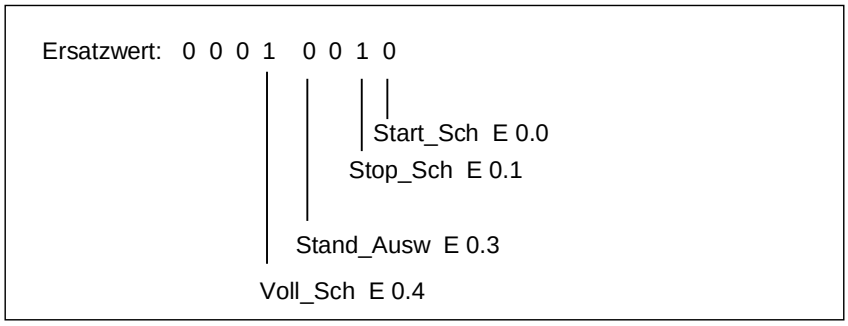
Bei allen Ladebefehlen, die zu synchronen Fehlern führen, können Sie im Fehler-OB einen Ersatzwert für den AKKU-Inhalt angeben.

Beispielprogramm zum Ersetzen eines Wertes

Im folgenden Beispielprogramm wird ein Ersatzwert in der SFC 44 RPL_VAL bereitgestellt. Nachfolgendes Bild zeigt, wie OB 122 aufgerufen werden kann, wenn die CPU erkennt, daß eine Eingabebaugruppe nicht reagiert.



In diesem Beispiel wird der Ersatzwert aus nachfolgendem Bild in das Programm eingegeben, damit das Programm mit sinnvollen Werten weiterlaufen kann.



Fällt eine Eingabebaugruppe aus, erzeugt die Bearbeitung der Anweisung L PEB0 einen synchronen Fehler und startet den OB 122. Standardmäßig liest der Ladebefehl den Wert 0 ein. Sie können jedoch mit der SFC 44 beliebige zum Prozeß passende Ersatzwerte definieren. Die SFC ersetzt den AKKU-Inhalt durch den vorgegebenen Ersatzwert.

Das folgende Beispielprogramm könnte in OB 122 gespeichert sein. Die folgende Tabelle zeigt die temporären Variablen, die in diesem Fall in der Variablendeklarationstabelle des OB 122 deklariert werden müssen.

Dekl.	Name	Typ	Beschreibung
TEMP	OB122_EV_CLASS	BYTE	Fehlerklasse/Fehlerkennung 29xx
TEMP	OB122_SW_FLT	BYTE	Fehlercode: 16#42, 16#43, 16#44 [1)], 16#45 [1)]
TEMP	OB122_PRIORITY	BYTE	Prioritätsklasse = Priorität des OB, in dem der Fehler aufgetreten ist
TEMP	OB122_OB_NUMBR	BYTE	122 = OB 122
TEMP	OB122_BLK_TYPE	BYTE	Bausteintyp, in dem der Fehler aufgetreten ist
TEMP	OB122_MEM_AREA	BYTE	Speicherbereich und Zugriffsart
TEMP	OB122_MEM_ADDR	WORD	Adresse im Speicher, an der der Fehler aufgetreten ist
TEMP	OB122_BLK_NUM	WORD	Nummer des Bausteins, in dem der Fehler aufgetreten ist
TEMP	OB122_PRG_ADDR	WORD	Relativadresse des fehlerverursachenden Befehls
TEMP	OB122_DATE_TIME	DATE_AND_TIME	Datum und Uhrzeit, zu denen der OB gestartet wurde
TEMP	Fehler	INT	Speichert den Fehlercode von SFC44

[1)] Nicht bei der S7-300.

AWL		Beschreibung
	L B#16#2942 L #OB122_SW_FLT ==I SPB QFeh L B#16#2943 <> I SPB Stop	Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2942) für die Quittierung eines Zeitfehlers beim Lesen von der Peripherie. Wenn gleich, springe zu “QFeh”. Vergleiche den Ereigniscode von OB 122 mit dem Ereigniscode (B#16#2943) für einen Adressierungsfehler (Schreiben von einer Baugruppe, die nicht vorhanden ist). Wenn ungleich, springe zu “Stop”.
Qfeh:	CALL ”REPL_VAL” VAL : = DW#16#2912 RET_VAL : = #Fehler L #Fehler L 0 ==I BEB	Marke ”QFeh”: Übergibt DW#16#2912 (binär 10010) an die SFC44 (REPL_VAL). SFC44 lädt diesen Wert in AKKU 1 (und ersetzt den Wert, der den Aufruf von OB 122 ausgelöst hat). Speichert den SFC-Fehlercode in #Fehler. Vergleicht #Fehler mit 0 (wenn gleich, dann ist bei der Bearbeitung von OB 122 kein Fehler aufgetreten). Beende den Baustein, wenn kein Fehler aufgetreten ist.
Stop:	CALL ”STP”	Marke ”Stop”: Ruft SFC46 ”STP” auf und versetzt die CPU in den Betriebszustand STOP.

21.9.5 Peripherie-Redundanzfehler (OB 70)

Beschreibung

Das Betriebssystem der H-CPU ruft den OB 70 auf, wenn ein Redundanzverlust am PROFIBUS DP auftritt (z. B. bei einem Busausfall am aktiven DP-Master oder bei einem Fehler in der Anschaltung des DP-Slave) oder wenn der aktive DP-Master von DP-Slaves mit geschalteter Peripherie wechselt

Programmieren des OB 70

Den OB 70 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 70 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 70 beispielsweise dazu nutzen:

- die Startinformation des OB 70 auszuwerten und festzustellen, welches Ereignis den Peripherie-Redundanzverlust ausgelöst hat.
- mit Hilfe der SFC 51 RDSYSST den Zustand Ihres Systems zu ermitteln (SZL_ID=B#16#71).

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein Peripherie-Redundanzfehler auftritt und der OB 70 nicht programmiert ist.

Ist der OB 70 geladen und das H-System befindet sich im redundanten Betrieb, so wird der OB 70 in beiden CPUs bearbeitet. Das H-System bleibt im redundanten Betrieb.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.6 CPU-Redundanzfehler (OB 72)

Beschreibung

Das Betriebssystem der H-CPU ruft den OB 72 auf, wenn eines der folgenden Ereignisse auftritt:

- Redundanzverlust der CPUs
- Vergleichsfehler (z. B. RAM, PAA)
- Reserve-Master-Umschaltung
- Synchronisationsfehler
- Fehler in einem SYNC-Modul
- Abbruch des Aufdatvorgangs
- Der OB 72 wird von allen CPUs ausgeführt, die sich nach einem zugehörigen Startereignis in den Betriebszuständen RUN oder ANLAUF befinden.

Programmieren des OB 72

Den OB 72 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 72 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 72 beispielsweise dazu nutzen,

- die Startinformation des OB 72 auszuwerten und festzustellen, welches Ereignis den CPU-Redundanzverlust ausgelöst hat
- mit Hilfe der SFC 51 RDSYSST den Zustand Ihres Systems zu ermitteln (SZL_ID=B#16#71)
- anlagenspezifisch auf den CPU-Redundanzverlust zu reagieren.

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein CPU-Redundanzfehler auftritt und der OB 72 nicht programmiert ist.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.7 Kommunikations-Redundanzfehler (OB 73)

Beschreibung

Das Betriebssystem der H-CPU ruft den OB 73 beim ersten Redundanzverlust einer hochverfügbaren S7-Verbindung auf (Hochverfügbare S7-Verbindungen gibt es ausschließlich bei der S7-Kommunikation, siehe "Automatisierungssystem S7-400 H Hochverfügbare Systeme"). Beim Redundanzverlust weiterer hochverfügbarer S7-Verbindungen erfolgt kein OB73-Start mehr.

Ein erneuter OB73-Start erfolgt erst wieder, wenn Sie bei allen S7-Verbindungen, die hochverfügbar waren, die Redundanz wieder hergestellt haben.

Die CPU wechselt nicht in den Betriebszustand STOP, wenn ein zugehöriges Startereignis auftritt und der OB 73 nicht programmiert ist.

Programmieren des OB 73

Den OB 73 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 73 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 73 beispielsweise dazu nutzen:

- die Startinformation des OB 73 auszuwerten und festzustellen, welches Ereignis den Peripherie-Redundanzverlust ausgelöst hat.
- mit Hilfe der SFC 51 RDSYSST den Zustand Ihres Systems zu ermitteln (SZL_ID=B#16#71).

Die CPU wechselt nicht in den Betriebszustand "STOP", wenn ein Kommunikations-Redundanzfehler auftritt und der OB 73 nicht programmiert ist.

Ist der OB 73 geladen und das H-System befindet sich im redundanten Betrieb, so wird der OB 73 in beiden CPUs bearbeitet. Das H-System bleibt im redundanten Betrieb.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.8 Zeitfehler (OB 80)

Beschreibung

Das Betriebssystem der CPU ruft OB 80 auf, wenn ein Zeitfehler auftritt. Zeitfehler können z. B. sein:

- Überschreiten der Maximalzykluszeit
- Überspringen von Uhrzeitalarmen durch Vorstellen der Uhrzeit
- zu große Verzögerung beim Bearbeiten einer Prioritätsklasse

Programmieren des OB 80

Den OB 80 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 80 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 80 beispielsweise dazu nutzen:

- die Startinformation des OB 80 auszuwerten und festzustellen, welche Uhrzeitalarme übersprungen wurden
- mit Hilfe der SFC 29 CAN_TINT den übersprungenen Uhrzeitalarm zu deaktivieren, damit er nicht ausgeführt wird und um mit der neu eingestellten Uhrzeit einen sauberen Aufsetzpunkt für die Bearbeitung der Uhrzeitalarme zu haben.

Wenn Sie übersprungene Uhrzeitalarme im OB 80 nicht deaktivieren, wird der erste übersprungene Uhrzeitalarm bearbeitet, alle anderen werden ignoriert.

Haben Sie OB 80 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Zeitfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.9 Stromversorgungsfehler (OB 81)

Beschreibung

Das Betriebssystem der CPU ruft OB 81 auf, wenn im Zentralgerät oder einem Erweiterungsgerät

- die 24V-Spannungsversorgung
- eine Batterie
- die komplette Pufferung

ausgefallen ist bzw. wenn die Störung beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 81

Den OB 81 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 81 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 81 beispielsweise dazu nutzen

- die Startinformation des OB 81 auszuwerten und festzustellen, welcher Stromversorgungsfehler vorliegt
- die Nummer des Baugruppenträgers mit der defekten Stromversorgung zu ermitteln
- eine Lampe an einer Operator Station anzusteuern, um dem Wartungspersonal anzuzeigen, daß eine Batterie auszutauschen ist.

Haben Sie OB 81 nicht programmiert, wechselt die CPU im Gegensatz zu allen anderen asynchronen Fehler-OBs nicht in den Betriebszustand "STOP", wenn ein Stromversorgungsfehler erkannt wird. Der Fehler wird aber in den Diagnosepuffer eingetragen und die entsprechende LED auf der Frontplatte zeigt den Fehler an.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.10 Diagnosealarm (OB 82)

Beschreibung

Das Betriebssystem der CPU ruft OB 82 auf, wenn eine diagnosefähige Baugruppe, bei der Sie den Diagnosealarm freigegeben haben, einen Fehler erkennt und wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 82

Den OB 82 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 82 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 82 beispielsweise dazu nutzen

- die Startinformation des OB 82 auszuwerten
- eine genaue Diagnose des aufgetretenen Fehlers vorzunehmen.

Wenn ein Diagnosealarm ausgelöst wird, trägt die gestörte Baugruppe automatisch 4 Byte Diagnosedaten und ihre Anfangsadresse in die Startinformation des Diagnosealarm-OBs und in den Diagnosepuffer ein. Damit erhalten Sie die Information, auf welcher Baugruppe, und wann ein Fehler aufgetreten ist.

Weitere Diagnosedaten der gestörten Baugruppe (auf welchem Kanal ist der Fehler aufgetreten, um welchen Fehler handelt es sich) können Sie mit einem entsprechenden Programm im OB 82 auswerten. Mit der SFC 51 RDSYSST können Sie Baugruppendiagnosedaten auslesen und mit der SFC 52 WR_USRMSG diese Information in den Diagnosepuffer eintragen. Außerdem können Sie zusätzlich die selbstdefinierte Diagnosemeldung an ein angemeldetes Beobachtungsgerät senden.

Haben Sie OB 82 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Diagnosealarm ausgelöst wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.11 Ziehen-/Stecken-Alarm (OB 83)

Beschreibung

S7-400-CPUs überwachen zyklisch im Abstand von ca. einer Sekunde das Ziehen und Stecken von Baugruppen im Zentralgerät und den Erweiterungsgeräten.

Nach dem Einschalten der Netzspannung prüft die CPU, ob alle Baugruppen, die in der mit STEP 7 erstellten Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind. Ist dies der Fall, wird diese Ist-Konfiguration gespeichert und dient als Referenzwert für die zyklische Überwachung der Baugruppen. In jedem Abfragezyklus wird die neu ermittelte Ist-Konfiguration mit der bisherigen Ist-Konfiguration verglichen. Bei Abweichungen wird ein Ziehen-/Stecken-Alarm gemeldet und es erfolgt je ein Eintrag in den Diagnosepuffer und die Systemzustandsliste. Im Betriebszustand RUN wird der Ziehen/Stecken-OB gestartet.

Hinweis

Im Betriebszustand RUN dürfen Stromversorgungsbaugruppen, CPUs und IMs nicht gezogen werden!

Zwischen Ziehen und Stecken einer Baugruppe müssen mindestens 2 Sekunden vergehe, damit das Ziehen oder Stecken von der CPU richtig erkannt werden kann.

Parametrierung einer neu gesteckten Baugruppe

Wird eine Baugruppe im Betriebszustand RUN gesteckt, prüft die CPU, ob der Baugruppentyp der neu gesteckten Baugruppe mit dem der ursprünglich gesteckten Baugruppe übereinstimmt. Bei Übereinstimmung der Baugruppentypen erfolgt die Parametrierung. Übertragen werden entweder die Default-Parameter oder die Parameter, die Sie mit STEP 7 vergeben haben.

Programmieren des OB 83

Den OB 83 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 83 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 83 beispielsweise dazu nutzen

- die Startinformation des OB 83 auszuwerten
- mit Hilfe der Systemfunktionen SFC 55 bis 59 die neu gesteckte Baugruppe nachzuparametrieren.

Haben Sie OB 83 nicht programmiert, wechselt die CPU von RUN in den Betriebszustand "STOP", wenn ein Ziehen-/Stecken-Alarm auftritt.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.12 CPU-Hardwarefehler (OB 84)

Beschreibung

Das Betriebssystem der CPU ruft OB 84 auf, wenn ein Fehler bei der Schnittstelle zum MPI-Netz, zum K-Bus oder zur Anschaltung für die Dezentrale Peripherie, z. B. fehlerhafter Signalpegel auf der Leitung erkannt wird bzw. wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 84

Den OB 84 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 84 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 84 beispielsweise dazu nutzen

- die Startinformation des OB 84 auszuwerten
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung zum Diagnosepuffer zu senden.

Haben Sie OB 84 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein CPU-Hardwarefehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.13 Programmablauffehler (OB 85)

Beschreibung

Das Betriebssystem der CPU ruft OB 85 auf, wenn ein

- Startereignis für einen Alarm-OB vorliegt, aber der OB nicht ausgeführt werden kann, weil er nicht in die CPU geladen wurde
- Fehler beim Zugriff auf den Instanz-Datenbaustein eines Systemfunktionsbausteins aufgetreten ist
- Fehler bei der Prozeßabbild-Aktualisierung (Baugruppe nicht vorhanden oder defekt) aufgetreten ist.

Programmieren des OB 85

Den OB 85 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 85 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 85 beispielsweise dazu nutzen

- die Startinformation des OB 85 auszuwerten und festzustellen, welche Baugruppe defekt ist oder fehlt (Angabe der Baugruppen-Anfangsadresse)
- mit Hilfe der SFC 49 LGC_GADR den Steckplatz der betreffenden Baugruppe zu ermitteln.

Haben Sie OB 85 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Prioritätsklassenfehler erkannt wird.

21.9.14 Baugruppenträgerausfall (OB 86)

Beschreibung

Das Betriebssystem der CPU ruft OB 86 auf, wenn ein Baugruppenträgerausfall erkannt wird, z. B. bei

- Baugruppenträgerausfall (fehlende oder defekte IM bzw. unterbrochene Verbindungsleitung
- dezentraler Spannungsausfall eines Baugruppenträgers
- Ausfall eines DP-Slaves in einem Mastersystem des Bussystems PROFIBUS-DP

bzw. wenn der Fehler beseitigt ist (Aufruf bei kommendem und gehendem Ereignis).

Programmieren des OB 86

Den OB 86 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 86 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 86 beispielsweise dazu nutzen

- die Startinformation des OB 86 auszuwerten und festzustellen, welcher Baugruppenträger defekt ist oder fehlt.
- mit Hilfe der Systemfunktion SFC 52 WR_USMSG eine Meldung in den Diagnosepuffer einzutragen und die Meldung zu einem Beobachtungsgerät zu senden.

Haben Sie OB 86 nicht programmiert, wechselt die CPU in den Betriebszustand STOP, wenn ein Baugruppenträgerausfall erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.15 Kommunikationsfehler (OB 87)

Beschreibung

Das Betriebssystem der CPU ruft OB 87 auf, wenn ein Kommunikationsfehler beim Datenaustausch über Kommunikations-Funktionsbausteine oder Globaldatenkommunikation auftritt, z. B.

- beim Empfang von Globaldaten wurde eine falsche Telegrammkennung erkannt
- der Datenbaustein für die Statusinformation der Globaldaten ist nicht vorhanden oder zu kurz.

Programmieren des OB 87

Den OB 87 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 87 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 87 beispielsweise dazu nutzen

- die Startinformation des OB 87 auszuwerten und
- wenn der Datenbaustein für die Statusinformation der Globaldatenkommunikation fehlt, einen Datenbaustein anzulegen.

Haben Sie OB 87 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Kommunikationsfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.16 Programmierfehler (OB 121)

Beschreibung

Das Betriebssystem der CPU ruft OB 121 auf, wenn ein Programmierfehler auftritt, z. B.

- adressierte Zeiten nicht vorhanden
- aufgerufener Baustein nicht geladen.

Programmieren des OB 121

Den OB 121 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 121 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 121 beispielsweise dazu nutzen

- die Startinformation des OB 121 auszuwerten
- die Fehlerursache in einen Melde-Datenbaustein einzutragen.

Haben Sie OB 121 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Programmierfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

21.9.17 Peripheriezugriffsfehler (OB 122)

Beschreibung

Das Betriebssystem der CPU ruft OB 122 auf, wenn mit einer STEP 7-Operation auf einen Ein- oder Ausgang einer Signalbaugruppe zugegriffen wird, dem zum Zeitpunkt des letzten Neustarts keine Baugruppe zugeordnet war, z. B.

- Fehler bei Peripherie-Direktzugriff (Baugruppe defekt oder nicht vorhanden)
- Zugriff auf eine Peripherie-Adresse, die der CPU nicht bekannt ist.

Programmieren des OB 122

Den OB 122 müssen Sie mit STEP 7 als Objekt in Ihrem S7-Programm erzeugen. Schreiben Sie das Programm, das im OB 122 bearbeitet werden soll, in den erzeugten Baustein und laden Sie ihn als Teil Ihres Anwenderprogramms in die CPU.

Sie können den OB 122 beispielsweise dazu nutzen

- die Startinformation des OB 122 auszuwerten
- die Systemfunktion SFC 44 aufzurufen und einen Ersatzwert für eine Eingabebaugruppe vorzugeben, damit das Programm mit einem sinnvollen, prozeßabhängigen Wert weiterbearbeitet werden kann.

Haben Sie OB 122 nicht programmiert, wechselt die CPU in den Betriebszustand "STOP", wenn ein Peripheriezugriffsfehler erkannt wird.

Ausführliche Informationen zu OBs, SFBs und SFCs finden Sie in den entsprechenden Baustein-Hilfen.

22 Drucken und Archivieren

22.1 Drucken einer Projektdokumentation

22.1.1 Drucken einer Projektdokumentation

Wenn Sie die Programmerstellung für Ihre Automatisierungslösung abgeschlossen haben, können Sie mit der in STEP 7 integrierten Druckfunktion alle wichtigen Daten ausdrucken und sich damit eine Projektdokumentation erzeugen.

Ausdruckbare Projektbestandteile

Den Inhalt von Objekten können Sie sowohl direkt vom SIMATIC Manager aus drucken als auch durch Öffnen des entsprechenden Objekts und anschließendem Druckanstoß.

Direkt über den SIMATIC Manager lassen sich folgende Bestandteile eines Projekts ausdrucken:

- Objektbaum (Struktur des Projekts/der Bibliothek)
- Objektlisten (Inhalt eines Objektordners)
- Objektinhalt (Inhalt eines Objekts)
- Meldungen

Über Öffnen des entsprechenden Objekts können z. B. folgende Projektbestandteile ausgedruckt werden:

- Bausteine in Darstellung KOP, FUP, AWL oder anderer Sprache (Optionssoftware)
- die Symboltabelle mit dem symbolischen Namen absoluter Adressen
- die Konfigurationstabelle mit der Anordnung der Baugruppen im AS und Baugruppenparametern
- den Diagnosepufferinhalt
- die Variablentabelle mit Statusformaten sowie Status- und Steuerwerten
- Referenzdaten, d.h. Querverweislisten, Belegungspläne, Programmstrukturen, nicht verwendete Operanden, Operanden ohne Symbol
- die Globaldatentabelle
- Auskunftsdaten zu Baugruppen mit Baugruppenzustand
- Bedienerrelevante Texte (Anwendertexte und Textbibliotheken)
- Dokumente von Optionspaketen z. B. von Programmiersprachen

Optionspaket DOCPRO

Zum Erzeugen, Bearbeiten und Drucken von standardisierten Schaltbüchern können Sie das Optionspaket DOCPRO einsetzen. Sie erhalten so eine Anlagendokumentation, die die DIN- und ANSI-Norm erfüllt.

22.1.2 Prinzipielle Vorgehensweise beim Drucken

Zum Ausdrucken gehen Sie folgendermaßen vor:

1. Öffnen Sie das geeignete Objekt, um die zu druckende Information am Bildschirm anzuzeigen.
2. Blenden Sie das Dialogfeld "Drucken" mit dem Menübefehl **Datei > Drucken...** im jeweiligen Fenster auf. Je nach Fenster kann der erste Eintrag in der Menüleiste anstelle von "Datei" auch anders lauten, z. B. "Tabelle".
3. Ändern Sie bei Bedarf die Druckeinstellungen (z. B. Drucker, Druckbereich, Anzahl der Kopien) im Dialogfeld und schließen Sie es.

Manche Dialogfelder enthalten die Schaltfläche "Drucken", z.B. das Dialogfeld zum Baugruppenzustand. Klicken Sie auf diese Schaltfläche, um den Inhalt des Dialogfelds auszudrucken.

Bausteine brauchen nicht geöffnet werden. Sie können diese direkt im SIMATIC Manager mit dem Menübefehl **Datei > Drucken**... ausgeben.

22.1.3 Funktionen zum Drucken

Für den Ausdruck der Druckobjekte stehen folgende ergänzende Funktionen zur Verfügung:

Druckobjekte	**Menübefehl**	**Funktion**	**Funktion**	**Funktion**	**Funktion**
		Druck-vorschau	Seite ein-richten	Schriftfelder	Drucker einrichten
Bausteine, AWL-Quellen	Datei > *	●	●	–	●
Baugrupppen-zustand		–	●	–	–
Globaldatentabelle	GD-Tabelle > *	●	●	–	●
Konfigurationstabelle	Station > *	●	●	–	●
Objekt, Objektordner	Datei > *	–	●	●	●
Referenzdaten	Referenzdaten > *	●	●	–	●
Symboltabelle	Tabelle > *	●	●	–	●
Variablentabelle	Tabelle > *	–	●	–	●
Verbindungstabelle	Netz > *	●	●	–	●
Bedienerrelevante Texte (Anwender-texte, Text-bibliotheken)	Texte > *	●	●	–	●

* : Das Zeichen * ist Platzhalter im Menübefehl für die jeweilige Funktion (z. B. Druckvorschau oder Seite einrichten)

Die Schritt-für-Schritt-Anleitungen, wie Sie zum Ausdrucken der einzelnen Druckobjekte vorgehen, finden Sie in der Online-Hilfe.

Druckvorschau

Über die Funktion "Druckvorschau" können Sie sich die Seitenansicht des Ausdrucks anzeigen lassen.

Wenn der Ausdruck aus mehreren Seiten besteht, wird nach der Seitennummer am rechten unteren Blattrand eine Folgeseitenkennung ausgegeben. Die letzte Seite enthält keine Folgeseitenkennung.

Hinweis

Das Druckbild des optimierten Ausdrucks wird in der Druckvorschau nicht angezeigt.

Seitenformat einstellen

Im SIMATIC Manager können Sie mit dem Menübefehl **Datei > Seite einrichten** das Seitenformat für alle STEP 7-Applikationen und Optionspakete einstellen (z. B. A4, A5, Letter). Temporär, das heißt für die aktuelle Sitzung, können in einigen Applikationen (z. B. Symbol Editor) andere Seitenformate eingestellt, jedoch nicht für weitere Sitzungen gespeichert, werden.

Passen Sie bitte das zum Ausdruck verwendete Formular dem gewünschten Papierformat an. Falls das Formular zu breit ist, wird der rechte Rand auf einem Folgeblatt ausgedruckt.

Wenn Sie ein Seitenformat mit Rand wählen (z.B. "A4 Rand"), erhält das Druckdokument an der linken Seite einen Rand, den Sie zum Lochen verwenden können.

Hinweis

Wenn Sie Hilfe zum Dialogfeld "Seite einrichten" benötigen, klicken Sie auf die Schaltfläche "Hilfe" oder drücken Sie die Taste F1, während sich der Cursor im Dialogfeld befindet.

Kopf- und Fußzeilen einstellen

Mit der Funktion "Datei > Schriftfelder" im SIMATIC Manager können Sie Kopf- und Fußzeilen für Ihre zu druckenden Dokumente projektweit einstellen. Umfaßt der Ausdruck mehrere Seiten, so wird am rechten unteren Blattrand eine Folgeseitenkennung ausgegeben. Die letzte Seite enthält keine Folgeseitenkennung. So sehen Sie schnell, ob der Ausdruck vollständig ist. Diese Folgeseitenkennung ist auch in der Druckvorschau sichtbar.

Drucker einrichten

Mit der Funktion "Drucker einrichten" können Sie einen Drucker auswählen und das Papierformat (Hochformat oder Querformat) einstellen. Die Einstellmöglichkeiten bei dieser Funktion hängen von dem verwendeten Druckertreiber ab.

22.1.4 Besonderheiten zum Drucken des Objektbaums

Im Dialogfeld "Objektliste drucken" können Sie durch Markieren des Optionsfelds "Baumfenster" zusätzlich zur Objektliste den Objektbaum ausdrucken.

Wenn Sie unter "Druckbereich" das Optionsfeld "Alles" markieren, so wird die gesamte Baumstruktur ausgedruckt. Bei Markieren des Auswahlknopfes "Auswahl" wird die Baumstruktur ab dem ausgewählten Objekt abwärts ausgedruckt.

Hinweis

Die in dem Dialogfeld gewählten Einstellungen gelten nur für das Drucken der Liste bzw. des Baumes, nicht jedoch für das Drucken der Inhalte. Hier werden die Einstellungen der jeweilig zuständigen Applikationen verwendet.

22.2 Archivieren von Projekten und Bibliotheken

22.2.1 Archivieren von Projekten und Bibliotheken

Einzelne Projekte bzw. Bibliotheken können Sie in komprimierter Form in einer Archivdatei ablegen. Dieses komprimierte Ablegen ist auf Festplatte oder transportablen Datenträgern (z.B. Diskette) möglich.

Archivier-Programme

Die Archivfunktion bietet Ihnen eine Schnittstelle für den Aufruf Ihres bevorzugten Archivier-Programms. Die Archivier-Programme ARJ und PKZIP 2.50 sind im Lieferumfang von STEP 7 enthalten. Um ein bestimmtes Archivier-Programm einzusetzen, benötigen Sie davon die folgende (oder eine neuere) Version:

- ARJ-Version ab 2.4.1a
- PKZIP-Version ab 2.04g
- LHARC-Version ab 2.13
- WinZip ab 6.0
- JAR-Version ab 1.02

Empfehlung für das Archivieren

Projekte, die “lange Dateinamen” (jenseits der 8.3-DOS-Konvention) oder tief geschachtelte Verzeichnisstrukturen (mit Verzeichnissen, deren absoluter Pfadname länger als 64 Zeichen ist) enthalten, sollen nur mit den Archivier-Programmen PKZIP 2.50, WinZip oder JAR archiviert werden. Bei den übrigen Archivier-Programmen ist für diese Strukturen nicht gewährleistet, daß sich die Archivdateien vollständig und korrekt entpacken lassen. Das gilt insbesondere für Projekte, die Objekte des Optionspakets WinCC enthalten.

22.2.2 Anwendungsfälle für Speichern / Archivieren

Speichern unter

Mit dieser Funktion können Sie eine **Kopie** des Projektes mit einem anderen Namen anlegen.

Diese Funktion läßt sich einsetzen

- zur Erstellung von Sicherungskopien,
- zur Duplizierung eines vorhandenen Projekts, um es an andere Anforderungen anzupassen.

Um möglichst schnell eine Kopie zu erstellen, wählen Sie im Folgedialog das Speichern ohne Reorganisation. Dann wird der gesamte Dateibaum vom Projektverzeichnis an ohne Überprüfung kopiert und unter einem neuen Namen gespeichert.

Auf dem Datenträger muß genügend Platz zur Ablage der Sicherungskopie vorhanden sein. Versuchen Sie nicht, Projekte auf Diskette zu speichern, da der Speicherplatz in der Regel nicht ausreichen wird. Verwenden Sie zum Transport von Projektdaten auf Disketten die Funktion "Archivieren".

Das Speichern mit Reorganisation dauert länger, es wird jedoch eine Meldung ausgegeben, wenn ein Objekt nicht kopiert und gespeichert werden kann. Ursachen können z.B. ein fehlendes Optionspaket oder defekte Daten eines Objekts sein.

Archivieren

Einzelne Projekte bzw. Bibliotheken können Sie in komprimierter Form in einer Archivdatei ablegen. Dieses komprimierte Ablegen ist auf Festplatte oder transportablen Datenträgern (z.B. Diskette) möglich.

Transportieren Sie Projekte auf Disketten nur in Form von Archivdateien. Falls das Projekt zu groß ist, wählen Sie ein Archivprogramm, mit dem diskettenübergreifende Archive erstellt werden können.

Projekte bzw. Bibliotheken, die in eine Archivdatei komprimiert wurden, können nicht bearbeitet werden. Wollen Sie diese erneut bearbeiten, müssen Sie die Daten entkomprimieren, d. h. das Projekt bzw. die Bibliothek dearchivieren.

22.2.3 Voraussetzungen für das Archivieren

Um ein Projekt/eine Bibliothek zu archivieren, müssen folgende Voraussetzungen erfüllt sein:

- Das Archivier-Programm muß auf Ihrem System installiert sein. Die Einbindung in STEP 7 wird in der Online-Hilfe im Abschnitt "Schritte zum Archivieren/Dearchivieren" erklärt.
- Alle Daten des Projekts müssen ausnahmslos in bzw. unterhalb des Projektverzeichnisses untergebracht sein. Beim Arbeiten mit der C-Entwicklungsumgebung ist es zwar möglich, Daten auch an anderer Stelle abzulegen. Diese würden aber dann nicht in die Archivdatei aufgenommen werden.
- Die Dateinamen müssen der DOS-Namenskonvention entsprechen (8 Zeichen zuzüglich 3 Zeichen für die Dateierweiterung), falls Sie mit einem der Archivier-Programme ARJ, PKZip Version 2.04g oder LHArc arbeiten. Denn diese Archivier-Programme sind DOS-Programme.
  PKZip Version 2.50, Jar und WinZip sind von dieser Einschränkung nicht betroffen.”.

22.2.4 Vorgehensweise beim Archivieren/Dearchivieren

Sie archivieren/dearchivieren Ihr Projekt/Ihre Bibliothek über den Menübefehl **Datei > Archivieren** bzw. **Datei > Dearchivieren**.

Hinweis

Projekte bzw. Bibliotheken, die in eine Archivdatei komprimiert wurden, können nicht bearbeitet werden. Wollen Sie diese erneut bearbeiten, müssen Sie die Daten entkomprimieren, d. h. das Projekt bzw. die Bibliothek dearchivieren.

Beim Dearchivieren werden die dearchivierten Projekte/Bibliotheken automatisch in die Projektliste/Bibliotheksliste aufgenommen.

Einstellen des Zielverzeichnisses

Um das Zielverzeichnis einzustellen, rufen Sie mit dem Menübefehl **Extras > Einstellungen** im SIMATIC Manager das Dialogfeld "Einstellungen" auf.

Im Register "Archivieren" dieses Dialogfelds können Sie die Option "Zielverzeichnis beim Dearchivieren abfragen" ein- und ausschalten.

Ist diese Option ausgeschaltet, wird beim Dearchivieren der Pfad als Zielpfad benutzt, der im Register "Allgemein" unter "Ablageort für Projekte" bzw. "Ablageort für Bibliotheken" dieses Dialogfelds angegeben ist.

Kopieren einer Archivdatei auf Disketten

Sie können ein Projekt bzw. eine Bibliothek archivieren und die Archivdatei dann auf eine Diskette kopieren. Es ist auch möglich, im Dialog ”Archivieren” gleich ein Diskettenlaufwerk auszuwählen.

Hinweis zum Dearchivieren mit PKZIP 2.04g

Falls beim Anlegen eines Archivs auf Diskette mit dem Archivier-Programm PKZIP die Option "Disketten-übergreifendes Archiv" aktiviert wurde, wird beim Dearchivieren verlangt, die letzte Diskette des Archivs einzulegen. Dazu gibt PKUNZIP im DOS-Fenster stets folgende Meldung aus:

```
Insert the LAST disk of the backup set - Press a key when
ready.
```

Diese Meldung erscheint auch dann, wenn das Archiv zwar mit der Option "Disketten-übergreifendes Archivieren" erstellt worden ist, das ganze Archiv aber auf eine einzige Diskette gepaßt hat.

Ignorieren Sie in diesem Fall die Meldung und quittieren Sie die Anzeige durch Drücken einer beliebigen Taste.

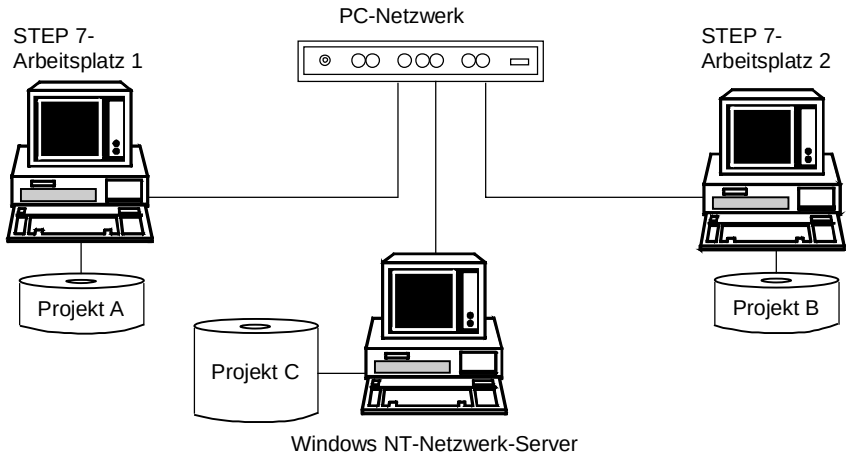
23 Mehrere Personen bearbeiten ein Projekt

23.1 Mehrbenutzer-Konfiguration im Windows-Netzwerkverbund

Übersicht

Mit STEP 7 können Sie in Windows 95/98/NT/2000-Workgroups und NT-/Novell-Netzwerken in einer Mehrbenutzer-Konfiguration arbeiten. Es sind prinzipiell drei Arbeitsweisen möglich:

- Das Projekt liegt auf einem lokalen Plattenlaufwerk und wird von einem anderen Arbeitsplatz mitbenutzt.
  Beispiel: Arbeitsplatz 1 und 2 greifen auf Projekt A am Arbeitsplatz 1 zu.
- Das Projekt liegt auf einem Projekt-/Netzwerk-Server.
  Beispiel: Arbeitsplatz 1 und 2 greifen auf Projekt C am Netzwerk-Server zu.
- Die Projekte liegen verteilt auf lokalen Plattenlaufwerken und auf einem oder mehreren Projekt-/Netzwerk-Servern.
  Beispiel: Arbeitsplatz 1 und 2 greifen auf die Projekte A, B und C zu.



Regeln zur Ablage von Projekten auf Netzwerk-Servern

- Wenn Sie Ihre Projekte auf Netzwerk-Servern ablegen, so muß der Zugriffspfad immer einem Laufwerksbuchstaben zugeordnet sein.
- Wenn Sie Ihre Projekte auf Netzwerk-Servern oder auf freigegebenen Laufwerken anderer Netzwerkteilnehmer ablegen, so darf an diesen Servern oder Netzwerkteilnehmern Windows 95/98/NT/2000 erst dann beendet werden, wenn alle STEP 7-Applikationen, die auf diese Projekte zugreifen, beendet sind.

Regeln zum Bearbeiten eines S7-Programm durch mehrere Personen

Beachten Sie folgendes:

- Bevor mehrere Personen an einem S7-Programm arbeiten, müssen Sie die Arbeitsplatz-Konfiguration einstellen (Menübefehl **Start > Simatic > STEP 7 > Simatic-Arbeitsplatz konfigurieren**).
- Bausteine bzw. AWL-Quelle:
  Jede Person sollte einen anderen Baustein bzw. eine andere AWL-Quelle programmieren. Versuchen zwei Personen gleichzeitig einen Baustein oder eine Quelle zu editieren, wird eine Meldung ausgegeben und der Zugriff für die zweite Person gesperrt.
- Symboltabelle:
  Mehrere Personen können gleichzeitig die Symboltabelle öffnen, aber nur ein Bearbeiter kann sie editieren. Versuchen zwei Personen gleichzeitig die Symboltabelle zu editieren, wird eine Meldung ausgegeben und der Zugriff für die zweite Person gesperrt.
- Variablentabelle:
  Mehrere Personen können gleichzeitig die Variablentabelle öffnen, aber nur ein Bearbeiter kann sie editieren. Versuchen zwei Personen gleichzeitig die Variablentabelle zu editieren, wird eine Meldung ausgegeben und der Zugriff für die zweite Person gesperrt. In einem S7-Programm kann es mehrere Variablentabellen geben. Diese können natürlich unabhängig voneinander bearbeitet werden.

Regeln zum Bearbeiten einer Station durch mehrere Personen

Beachten Sie folgendes:

- Die Hardwarekonfiguration und die Netzkonfiguration einer Station sollten nur zentral von einer Person bearbeitet werden.

24 Arbeiten mit M7-Automatisierungssystemen

24.1 Vorgehensweise für M7-Systeme

Durch seine standardisierte PC-Architektur bildet der Automatisierungsrechner M7-300/400 eine freiprogrammierbare Erweiterung der SIMATIC-Automatisierungsplattform. Die Anwenderprogramme für SIMATIC M7 können Sie in einer Hochsprache wie C oder auch grafisch mit CFC (Continuous Funktion Chart) programmieren.

Für die Erstellung der Programme benötigen Sie zusätzlich zu STEP 7 die Systemsoftware M7-SYS RT für M7-300/400 sowie eine Entwicklungsumgebung für M7-Programme (ProC/C++ oder CFC).

Prinzipielles Vorgehen

Bei der Erstellung einer Automatisierungslösung mit SIMATIC M7 fallen grundlegende Aufgaben an. Die folgende Tabelle zeigt die Aufgaben, die in den meisten Projekten durchgeführt werden, und ordnet sie zu einer prinzipiellen Vorgehensweise in Form eines Leitfadens. Sie verweist dabei auf die jeweiligen Kapitel dieses Handbuchs oder weiterer Handbücher.

Vorgehen	**Beschreibung**
Automatisierungslösung entwerfen	M7 spezifisch; siehe: PHB M7-SYS RT
STEP 7 starten	wie bei STEP 7
Projektstruktur anlegen Station einrichten Hardware konfigurieren	wie bei STEP 7
Kommunikationsverbindungen projektieren	wie bei STEP 7
Symboltabelle definieren	wie bei STEP 7
C- oder CFC-Anwenderprogramm erstellen	M7 spezifisch; siehe: ProC/C++
Betriebssystem konfigurieren Betriebssystem auf M7-300/400 installieren Hardwarekonfiguration und Anwenderprogramm in M7 laden	M7 spezifisch; siehe: BHB M7-SYS Rt
Anwenderprogramm testen und debuggen	ProC/C++
Betrieb überwachen und M7 diagnostizieren	wie bei STEP 7, jedoch ohne anwenderdefinierte Diagnose
Drucken und Archivieren	wie bei STEP 7

Was ist anders?

Für M7-300/400 werden in STEP 7 folgende Funktionen nicht unterstützt:

- Multicomputing - Synchroner Betrieb mehrerer CPUs
- Forcen von Variablen
- Globaldaten-Kommunikation
- Anwenderdefinierte Diagnose

Verwalten der M7-Zielsysteme

STEP 7 unterstützt Sie speziell bei folgenden Arbeiten mit M7-Automatisierungssystemen:

- Betriebssystem auf dem M7-300/400 installieren
- Konfigurierung des Betriebssystems durch Editieren von Systemdateien
- Übertragen von Anwenderprogrammen in den M7-300/400
- Firmware aktualisieren

Um in die M7-Zielsystemverwaltung zu gelangen, rufen Sie aus dem Kontext eines Projektes, das Stationen mit M7-CPUs oder -FMs enthält, mit angewähltem M7-Programmordner, folgenden Menübefehl auf:

Zielsystem > M7-Zielsystem verwalten

Genaue Anleitungen finden Sie in der Online-Hilfe und im Benutzerhandbuch M7-SYS RT.

24.2 Optionssoftware für M7-Programmierung

M7-Optionssoftware

STEP 7 bietet Ihnen die Grundfunktionalität an, die Sie benötigen um:

- Projekte zu erzeugen und zu verwalten
- Hardware zu konfigurieren und zu parametrieren
- Netzwerke und Verbindungen zu projektieren
- Symboldaten zu verwalten

Diese Funktionalität ist unabhängig davon, ob Ihr Automatisierungssystem SIMATIC S7 oder SIMATIC M7 ist.

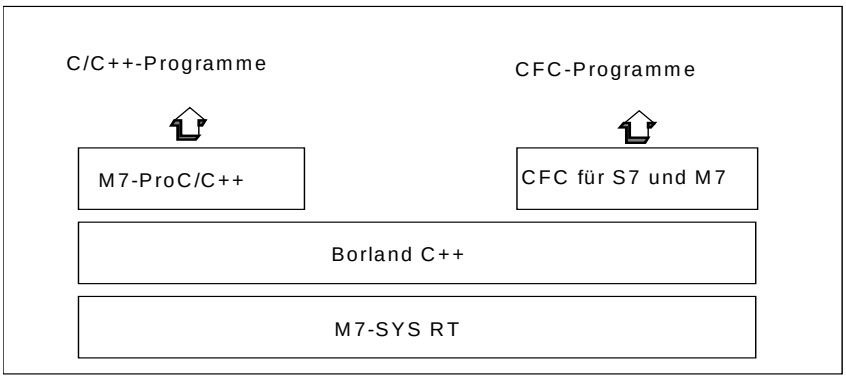
Für die Erstellung von M7-Anwenderprogrammen benötigen Sie zusätzlich zu STEP 7 noch M7-Optionssoftware:

Software	Inhalt
M7-SYS RT	• Betriebssystem M7 RMOS32 • Systembibliothek M7-API • MPI-Unterstützung
CFC für S7 und M7	Programmiersoftware für CFC-Anwendungen (CFC = Continuous Function Chart)
M7-ProC/C++	• Einbindung der Borland-Entwicklungsumgebung in STEP 7 • Symbolimport-Editor und -Generator • Hochsprachen-Debugger Organon xdb386
Borland C++	Borland C++-Entwicklungsumgebung

Zusammen mit der M7-Optionssoftware unterstützt Sie STEP 7 zusätzlich bei folgenden Tätigkeiten:

- Daten auf das den M7 über MPI übertragen
- Informationen über das M7-Automatisierungssystem abfragen
- bestimmte Einstellungen auf dem M7-Automatisierungssystem vornehmen und den M7 urlöschen

Das folgende Bild zeigt die Abhängigkeiten der M7-Optionssoftware für die M7-Programmierung.



Zusammenfassung

Für die Erstellung von ...	benötigen Sie die M7-Optionssoftware ...
C/C++-Programmen	• M7-SYS RT • M7-ProC/C++ • Borland C++
CFC-Programmen	• M7-SYS RT • CFC für S7 und M7 • Borland C++

Wo Sie Unterstützung finden

Die spezifischen Werkzeuge für die Erstellung von M7-Anwendungen sind teilweise in STEP 7 und teilweise in der M7-Optionssoftware integriert.

Aus der folgenden Tabelle erfahren Sie, welche Unterstützung Sie in den einzelnen Softwarepaketen finden:

Software ...	Unterstützt Sie ...
STEP 7	• beim Installieren des M7-Betriebssystems, • der Verwaltung des M7-Automatisierungssystems, • dem Übertragen, Starten und Löschen der M7-Anwender-programme • dem Abrufen von Zustands- und Diagnosedaten • beim Urlöschen der CPU
M7-SYS RT	durch die Dienste des M7-Betriebssystems und der Systemsoftware bei: • Programmablaufsteuerung, • Speicher- und Ressourcenverwaltung, • Zugriff auf Rechner- und SIMATIC-Hardware, • Alarmverwaltung, • Diagnose, • Zustandsüberwachung und • Kommunikation
M7-ProC/C++	• durch integrierte Code-Erzeugung (Integration der Borland-Entwicklungsumgebung in STEP 7) • durch Einbindung der Projekt-Symbolik in den Quellcode und • durch integrierte Debug-Funktionalität
Borland C++	• beim Erstellen von C- und C++-Programmen
CFC für S7 und M7	• beim Erstellen, Testen und Debuggen von CFC-Programmen und • beim Starten und Ablaufen von CFC-Programmen

24.3 Betriebssysteme für M7-300/400

Für Anwendungen, die in den Hochsprachen C und C++ erstellt werden, sind die Dienste des Betriebssystems von entscheidender Bedeutung. Das Betriebssystem übernimmt für die Anwendung folgende Aufgaben:

- Zugriff auf die Hardware
- Verwaltung der Ressourcen
- Systemeinbindung
- Kommunikation zu anderen Komponenten im System

Für die Lösung von Automatisierungsaufgaben setzen wir auf dem Automatisierungsrechner SIMATIC M7 das Echtzeitbetriebssystem M7 RMOS32 (RMOS = **R**ealtime-**M**ultitasking-**O**perating-**S**ystem) ein. Für die Einbindung in das SIMATIC-System ist M7 RMOS32 um eine Aufrufschnittstelle, das M7-API (API = **A**pplication **P**rogramming **I**nterface) erweitert.

Das Echtzeitbetriebssystem M7 RMOS32 wird für 32-Bit-Anwendungen zur Lösung zeitkritischer Echtzeit- und Multitasking-Aufgaben eingesetzt. Es ist für M7-Baugruppen in folgenden Konfigurationen verfügbar:

- M7 RMOS32
- M7 RMOS32 mit MS-DOS

Welche Betriebssystem-Konfiguration Sie für Ihr M7-Automatisierungssystem auswählen hängt von den M7-Baugruppen, die Sie einsetzen, ab:

Betriebssystem-Konfiguration	**Baugruppe / Hauptspeicher**	**PROFIBUS-DP und TCP/IP: Ja/Nein**	**Installation auf Massenspeicher**
M7 RMOS32	FM 356-4 / 4MB	nein	Memory Card ≥4 MB oder Festplatte
	FM 356-4 / 8MB	ja	
	CPU 388-4 / 8MB	ja	
	FM 456-4 / 16MB	ja	
	CPU 488-3 / 16MB	ja	
	CPU 486-3 / 16MB	ja	
M7 RMOS32 mit MS-DOS	FM 356-4 / 8MB	nein	Memory Card ≥4 MB oder Festplatte
	CPU 388-4 / 8MB	nein	
	FM 456-4 / 16MB	ja	
	CPU 488-3 / 16MB	ja	
	CPU 486-3 / 16MB	ja	

25 Tips und Tricks

25.1 Baugruppen tauschen in der Konfigurationstabelle

Wenn Sie eine Stationskonfiguration mit HW Konfig überarbeiten und eine Baugruppen tauschen möchten, z. B. gegen eine mit neuer Bestell-Nr., dann gehen Sie folgendermaßen vor:

1. Ziehen Sie die Baugruppe per Drag&Drop aus dem Fenster "Hardware Katalog" über die bereits plazierte ("alte") Baugruppe.
2. Lassen Sie die neue Baugruppe "fallen"; die neue Baugruppe übernimmt, soweit möglich, die Parameter von der bereits gesteckten.

Diese Vorgehensweise ist schneller als das Austauschen durch Löschen der alten und anschließendem Einfügen und Parametrieren der neuen Baugruppe.

Diese Funktion können Sie über den Menübefehl **Extras > Einstellungen** ("Baugruppen austauschen ermöglichen") in HW Konfig gezielt ein- bzw. ausschalten.

25.2 Projekte mit einer großen Anzahl vernetzter Stationen

Wenn Sie nacheinander alle Stationen konfigurieren und dann über **Extras > Netz konfigurieren** NetPro aufrufen, um Verbindungen zu projektieren, dann werden die Stationen automatisch in der Netzsicht plaziert. Das hat den Nachteil, daß Sie die Stationen und Subnetze nachträglich nach topologischen Kriterien ordnen müssen.

Falls Ihr Projekt eine große Anzahl vernetzter Stationen umfaßt und Sie Verbindungen zwischen diesen Stationen projektieren wollen, dann sollten Sie von Anfang an die Anlagenstruktur in der Netzsicht projektieren, um die Übersicht zu bewahren:

1. Erzeugen Sie das neue Projekt im SIMATIC Manager (Menübefehl **Datei > Neu**)
2. Starten Sie NetPro (Menübefehl **Extras > Netz konfigurieren**)

3. Erzeugen Sie in NetPro Station für Station:
   - Plazieren Sie die Station per Drag&Drop aus dem Fenster "Katalog"
   - Doppelklicken Sie auf die Station, um HW Konfig zu starten
   - Plazieren Sie in HW Konfig per Drag&Drop die kommunikationsfähigen Baugruppen (CPUs, CPs, FMs, IF-Module)
   - Falls Sie diese Baugruppen vernetzen wollen: Doppelklicken Sie auf die entsrechenden Zeilen in der Konfigurationstabelle, um Subnetze neu anzulegen und die Schnittstellen zu vernetzen
   - Speichern Sie die Konfiguration und wechseln Sie zu NetPro
   - Positionieren Sie in NetPro Stationen und Subnetze (Objekt mit Mauszeigen verschieben, bis die gewünschte Position erreicht ist)
4. Projektieren Sie in NetPro die Verbindungen und korrigieren Sie ggf. die Vernetzung

25.3 Reorganisation

Falls unerklärliche Probleme beim Arbeiten mit STEP 7 auftreten, hilft es oft, die Datenhaltung des Projekts oder der Bibliothek zu reorganisieren.

Dazu wählen Sie den Menübefehl **Datei > Reorganisieren**. Bei der Reorganisation werden die durch Löschen entstandenen Lücken beseitigt, d.h. der Speicherbedarf der Projekt-/Bibliotheksdaten wird verkleinert.

Die Funktion optimiert die Datenablage für das Projekt bzw. die Bibliothek ähnlich wie z.B. ein Programm zum Defragmentieren einer Festplatte die Datenablage auf der Festplatte optimiert.

Die Dauer des Reorganisierens hängt von den erforderlichen Datenbewegungen ab und kann längere Zeit in Anspruch nehmen. Die Funktion wird deshalb nicht automatisch (z.B. beim Schließen eines Projekts) durchgeführt, sondern muß vom Benutzer gestartet werden, wenn er das Projekt oder die Bibliothek reorganisieren möchte.

Voraussetzung

Projekte und Bibliotheken können nur dann reorganisiert werden, wenn keine Objekte daraus von anderen Applikationen bearbeitet werden und somit gesperrt sind.

25.4 Testen mit der Variablentabelle

Für das Beobachten und Steuern von Variablen in der Variablentabelle gibt es eine Reihe von Bearbeitungstips:

- Symbole und Operanden können Sie sowohl in die Spalte "Symbol" als auch in die Spalte "Operand" eintragen. Der Eintrag wird automatisch in die passende Spalte geschrieben.
- Damit Ihnen der gesteuerte Wert angezeigt wird, sollten Sie den Triggerpunkt für "Beobachten" auf "Beginn Zyklus" und den Triggerpunkt für "Steuern" auf "Ende Zyklus" einstellen.
- Wenn Sie den Cursor in eine rot markierte Zeile plazieren, können Sie mit dem Tooltip die Ursache des Fehlers ablesen. Mit der Taste F1 erhalten Sie Hinweise zur Behebung.
- Sie können nur solche Symbole eintragen, die bereits in der Symboltabelle definiert sind. Ein Symbol muß genau so eingegeben werden, wie es in der Symboltabelle definiert ist. Symbolnamen, die Sonderzeichen enthalten, müssen in Hochkommas eingeschlossen werden (z.B. "Motor.Aus", "Motor+Aus", "Motor-Aus").
- Ausschalten von Warnhinweisen im Register "Online" (Dialogfeld "Einstellungen").
- Wechseln der Verbindung ist möglich, ohne vorher die Verbindung abgebaut zu haben.
- Der Trigger für Beobachten ist während des Beobachtens von Variablen einstellbar.
- Sie können einzelne ausgewählte Variablen steuern, indem Sie die betreffenden Zeilen markieren und die Funktion "Steuern" ausführen. Nur die sichtbaren markierten Variablen werden dann gesteuert.
- Mit der Option "Variablen zusammenfassen" können Sie die Anzahl der max. beobachtbaren Variablen vergrößern (Register "Online"im Dialogfeld "Einstellungen").
- Beenden ohne Rückfrage: Wenn Sie - bei laufendem "Beobachten", "Steuern" "PA-freischalten"- die Taste ESC drücken, werden "Beobachten" und "Steuern" ohne Rückfrage beendet.
- Eingeben eines zusammenhängenden Operandenbereichs:
  Verwenden Sie den Menübefehl **Einfügen > Bereich**.
- Ein- und Ausblenden von Spalten:
  Verwenden Sie die folgenden Menübefehle, um einzelne Spalten ein- oder auszublenden:
  Symbol: Menübefehl **Ansicht > Symbol**
  Symbolkommentar: Menübefehl **Ansicht > Symbolkommentar**
  Darstellungsformat des Statuswertes: Menübefehl **Ansicht > Anzeigeformat**
  Statuswert der Variablen: Menübefehl **Ansicht > Statuswert**
  Steuerwert der Variablen: Menübefehl **Ansicht > Steuerwert**
- Ändern des Anzeigeformats mehrerer Tabellenzeilen gleichzeitig:
  - Markieren Sie den Bereich der Tabelle, in dem Sie das Anzeigeformat ändern wollen, indem Sie mit gedrückter linker Maustaste über den gewünschten Tabellenbereich ziehen.

  - Wählen Sie die Darstellung mit dem Menübefehl **Ansicht > Anzeigeformat wählen**. Es wird nur für die Zeilen der markierten Tabelle das Format gewechselt, für die der Formatwechsel erlaubt ist.

- Eingabebeispiele über die Taste F1:
  - Wenn Sie den Cursor in die Operandenspalte plazieren und F1 drücken, erhalten Sie Beispiele für die Eingabe von Operanden.
  - Wenn Sie den Cursor in die Steuerwertspalte plazieren und F1 drücken, erhalten Sie Beispiele für die Eingabe von Steuer-/Forcewerten.

25.5 Virtueller Arbeitsspeicher

Ein weiterer Grund für Fehlverhalten von STEP 7 kann ein zu kleiner virtueller Arbeitsspeicher sein.

Zum Arbeiten mit STEP 7 unter Windows 95/98/NT/2000 sollten Sie die Einstellung für den virtuellen Arbeitsspeicher anpassen. Gehen Sie folgendermaßen vor:

1. Öffnen Sie die Systemsteuerung z.B. aus der Startleiste mit **Start > Einstellungen > Systemsteuerung**.
2. Doppelklicken Sie auf das Symbol "System".
3. Wählen Sie unter Windows 95/98/NT im angezeigten Dialogfeld das Register "Leistungsdaten" / "Leistungsmerkmale".
   Wählen Sie unter Windows 2000 im angezeigten Dialogfeld das Register "Erweitert" und betätigen Sie die Schaltfläche "Systemleistungsoptionen".
4. Betätigen Sie die Schaltfläche "Virtueller Arbeitsspeicher" (bei Windows 9x) bzw. "Ändern" (bei Windows NT/2000).
5. Nur 9x: Wählen Sie im Dialogfeld "Virtueller Arbeitsspeicher" die Option "Es gelten benutzerdefinierte Einstellungen für den virtuellen Speicher.
6. Geben Sie als "Minimum" bzw. "Anfangsgröße" mindestens 40 MByte und als "Maximum" bzw. "Maximalgröße" mindestens 150 MByte an.
7. Nur 9x: Stellen Sie sicher, daß die Option "Virtuellen Speicher deaktivieren" ausgeschaltet ist.
   Nur NT: Betätigen Sie die Schaltfläche "Setzen".

Hinweis

Da sich der virtuelle Arbeitsspeicher auf der Platte befindet (Default C: und dynamisch) soll darauf geachtet werden, daß für das Verzeichnis TMP bzw. TEMP ausreichend Platz zur Verfügung steht (ca. 20 - 30 MB):

- Sollte sich das S7-Projekt ebenfalls auf der selben Partition befinden, auf dem der virtuelle Arbeitsspeicher eingestellt wurde, so sollte ca. 2facher Speicherbedarf des S7-Projekts bereitgehalten werden.
- Wenn das Projekt jedoch auf einer anderen Partition verwaltet wird, so entfällt diese Bedingung.

A Anhang

A.1 Betriebszustände

A.1.1 Betriebszustände und Übergänge

Betriebszustände

Betriebszustände beschreiben das Verhalten der CPU zu jedem beliebigen Zeitpunkt. Das Wissen über die Betriebszustände der CPUs ist nützlich für die Programmierung des Anlaufs, den Test der Steuerung sowie für die Fehlerdiagnose.

Die S7-300- und S7-400-CPUs können folgende Betriebszustände annehmen:

- STOP
- ANLAUF
- RUN
- HALT

Im Betriebszustand "STOP" prüft die CPU ob alle konfigurierten bzw. über Default-Adressierung eingesetzten Baugruppen vorhanden sind und setzt die Peripherie in einen vordefinierten Grundzustand. Das Anwenderprogramm wird im Betriebszustand "STOP" nicht bearbeitet.

Im Betriebszustand "ANLAUF" unterscheidet man zwischen den Anlaufarten "Neustart" (Warmstart), "Kaltstart" und Wiederanlauf:

- Bei Neustart (Warmstart) wird die Programmbearbeitung am Programmanfang mit einer ”Grundstellung” der Systemdaten und der Anwenderoperandenbereiche neu begonnen (die nicht remanenten Zeiten, Zähler und Merker werden zurückgesetzt).
- Bei Kaltstart wird das Prozeßabbild der Eingänge eingelesen und das STEP 7-Anwenderprogramm beginnend beim ersten Befehl im OB1 bearbeitet (gilt auch für Neustart (Warmstart)).
  - Per SFC erzeugt Datenbausteine im Arbeitsspeicher werden gelöscht, die übrigen Datenbausteine haben den vorbelegten Wert aus dem Ladespeicher
  - Das Prozeßabbild sowie alle Zeiten, Zähler und Merker werden zurückgesetzt, unabhängig davon, ob sie als remanent parametriert worden sind.

- Bei Wiederanlauf wird die Programmbearbeitung an der unterbrochenen Stelle fortgesetzt (Zeiten, Zähler und Merker werden nicht zurückgesetzt). Wiederanlauf ist nur bei S7-400-CPUs möglich.

Im Betriebszustand "RUN" bearbeitet die CPU das Anwenderprogramm, aktualisiert die Ein- und Ausgänge, bearbeitet Alarme und Fehlermeldungen.

Im Betriebszustand "HALT" wird die Bearbeitung des Anwenderprogramms angehalten, und Sie können das Anwenderprogramm schrittweise testen. Der Betriebszustand "HALT" kann nur beim Testen mit dem PG erreicht werden.

In all diesen Betriebszuständen ist die CPU über die MPI-Schnittstelle kommunikationsfähig.

Weitere Betriebszustände

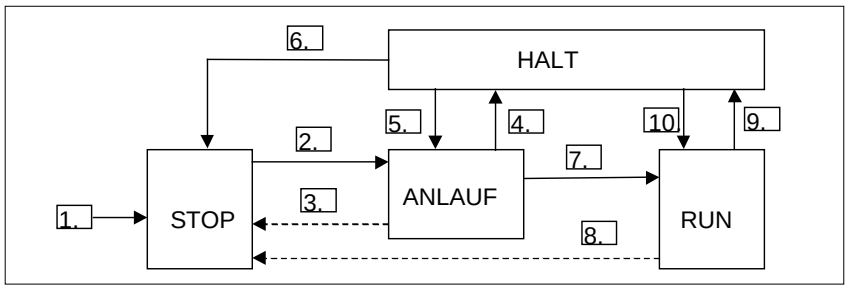
Ist die CPU nicht betriebsbereit, befindet sie sich in einem der beiden Betriebszustände:

- Spannungslos, d.h. die Netzspannung ist ausgeschaltet.
- Defekt, d.h. es ist ein nicht behebbarer Fehler aufgetreten.

  Überprüfen Sie, ob die CPU tatsächlich defekt ist: Setzen Sie die CPU auf STOP, und schalten Sie den Netzschalter aus und wieder ein. Wenn die CPU anläuft, lesen Sie den Diagnosepuffer aus, um den Fehler zu analysieren. Wenn die CPU nicht anläuft, muß sie ausgetauscht werden.

Betriebszustandsübergänge

Nachfolgendes Bild zeigt die Betriebszustände und die Betriebszustandsübergänge der S7-300- und S7-400-CPUs:



Die Bedingungen, unter welchen die Betriebszustände wechseln, können Sie folgender Tabelle entnehmen.

Übergang	Beschreibung
1.	Nach dem Einschalten der Versorgungsspannung befindet sich die CPU im Betriebszustand "STOP".
2.	Die CPU geht in den Betriebszustand "ANLAUF" • nachdem die CPU über Schlüsselschalter oder vom PG aus in RUN oder RUN-P gebracht wurde oder • nach automatischer Auslösung einer Anlaufart durch NETZ-EIN. • Wenn die Kommunikationsfunktion "RESUME" oder "START" ausgeführt wird. Der Schlüsselschalter muß in beiden Fällen auf RUN oder RUN-P stehen.
3.	Die CPU geht wieder in den STOP-Zustand, wenn • während des Anlaufs ein Fehler erkannt wird • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird • ein Stopp-Befehl im Anlauf-OB bearbeitet wird oder • die Kommunikationsfunktion "STOP" ausgeführt wird.
4.	Die CPU geht in den Betriebszustand "HALT", wenn im Anlaufprogramm ein Haltepunkt erreicht wird.
5.	Die CPU geht in den Betriebszustand "ANLAUF", wenn der Haltepunkt in einem Anlaufprogramm gesetzt war und der Befehl ”HALT VERLASSEN” ausgeführt wird (Testfunktion).
6.	Die CPU geht wieder in den STOP-Zustand, wenn • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird oder • der Kommunikationsbefehl "STOP" ausgeführt wird.
7.	Ist der Anlauf erfolgreich, dann wechselt die CPU in RUN.
8.	Die CPU geht wieder in den STOP-Zustand, wenn • im RUN-Zustand ein Fehler erkannt wird und der zugehörige OB nicht geladen ist • die CPU über Schlüsselschalter oder vom PG aus auf STOP gesetzt wird • ein Stopp-Befehl im Anwenderprogramm bearbeitet wird oder • die Kommunikationsfunktion "STOP" ausgeführt wird.
9.	Die CPU geht in den Betriebszustand "HALT", wenn im Anwenderprogramm ein Haltepunkt erreicht wird.
10.	Die CPU geht in den Betriebszustand "RUN", wenn ein Haltepunkt gesetzt war und der Befehl ”HALT VERLASSEN” ausgeführt wird.

Priorität der Betriebszustände

Werden mehrere Betriebszustandswechsel gleichzeitig angefordert, wird in den Betriebszustand mit der höchsten Priorität gewechselt. Steht z.B. der Betriebsartenschalter auf RUN, und vom PG aus wird versucht, die CPU in STOP zu schalten, geht die CPU in STOP, weil dieser Betriebszustand die höchste Priorität hat.

Priorität	Betriebszustand
Höchste	STOP
	HALT
	ANLAUF
Niedrigste	RUN

A.1.2 Betriebszustand STOP

Im Betriebszustand "STOP" wird das Anwenderprogramm nicht bearbeitet. Alle Ausgänge werden auf Ersatzwerte gesetzt und damit der gesteuerte Prozeß in einen sicheren Betriebszustand gebracht. Die CPU prüft, ob

- Hardwareprobleme vorliegen (z. B. Baugruppen nicht verfügbar sind)
- für die CPU die Defaulteinstellung gelten soll oder Parametersätze vorliegen
- die Randbedingungen für das programmierte Anlaufverhalten stimmen
- Systemsoftwareprobleme vorliegen.

Im STOP-Zustand können auch Globaldaten empfangen werden, und es kann passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden.

Urlöschen

Im STOP-Zustand kann die CPU urgelöscht werden. Urlöschen kann manuell über den Schlüsselschalter (MRES) oder vom PG aus (z.B. vor dem Laden eines Anwenderprogramms) erfolgen.

Durch Urlöschen wird die CPU in den ”Urzustand” versetzt, d.h.

- das gesamte Anwenderprogramm im Arbeitsspeicher und im RAM-Ladespeicher sowie alle Operandenbereiche werden gelöscht.
- die Systemparameter sowie die CPU- und Baugruppenparameter werden auf die Defaulteinstellung zurückgesetzt. Die vor dem Urlöschen eingestellten MPI-Parameter bleiben erhalten.
- wenn eine Memory Card (Flash-EPROM) gesteckt ist, kopiert die CPU das Anwenderprogramm aus der Memory Card in den Arbeitsspeicher (inkl. CPU- und Baugruppenparameter, falls die entsprechenden Konfigurationsdaten sich ebenfalls auf der Memory Card befinden).

Der Diagnosepuffer, die MPI-Parameter, die Uhrzeit und der Betriebsstundenzähler werden nicht zurückgesetzt.

A.1.3 Betriebszustand ANLAUF

Bevor die CPU nach dem Einschalten mit der Bearbeitung des Anwenderprogramms beginnt, wird ein Anlaufprogramm bearbeitet. Im Anlaufprogramm können Sie durch entsprechende Programmierung von Anlauf-OBs bestimmte Voreinstellungen für Ihr zyklisches Programm festlegen.

Es gibt drei Anlaufarten: Neustart (Warmstart), Kaltstart und Wiederanlauf. Ein Wiederanlauf ist grundsätzlich nur bei S7-400-CPUs möglich. Er muß im Parametersatz der CPU mit STEP 7 festgelegt worden sein.

Im Betriebszustand "ANLAUF":

- wird das Programm im Anlauf-OB (OB 100 für Neustart (Warmstart), OB 101 für Wiederanlauf, OB 102 für Kaltstart) abgearbeitet
- ist keine zeit- und alarmgesteuerte Programmbearbeitung möglich
- werden die Zeiten aktualisiert
- läuft der Betriebsstundenzähler
- sind die Digitalausgänge auf Signalbaugruppen gesperrt, können aber über Direktzugriff gesetzt werden.

Neustart (Warmstart)

Ein Neustart (Warmstart) ist immer zulässig, es sei denn, vom System wurde Urlöschen angefordert. In folgenden Fällen ist nur Neustart (Warmstart) möglich, nach:

- Urlöschen
- dem Laden des Anwenderprogramms im STOP-Zustand der CPU
- USTACK/BSTACK-Überlauf
- Abbruch des Neustart (Warmstart) (durch NETZ-AUS oder über Betriebsartenschalter)
- Überschreiten der parametrierten Unterbrechungszeitgrenze für Wiederanlauf.

Manueller Neustart (Warmstart)

Ein manueller Neustart (Warmstart) kann ausgelöst werden:

- über den Betriebsartenschalter

  (der Schalter CRST/WRST – soweit vorhanden - muß auf CRST stehen)
- über Menübefehl vom PG aus bzw. über Kommunikationsfunktionen

  (wenn der Betriebsartenschalter auf RUN oder RUN-P steht)

Automatischer Neustart (Warmstart)

Ein automatischer Neustart (Warmstart) kann ausgelöst werden bei NETZ-EIN, wenn:

- die CPU bei NETZ-AUS nicht im STOP war
- der Betriebsartenschalter auf RUN oder RUN-P steht
- kein automatischer Wiederanlauf nach NETZ-EIN parametriert ist
- die CPU im Neustart (Warmstart) durch Netzausfall unterbrochen wurde (unabhängig von der Parametrierung der Anlaufart).

Der Schalter CRST/WRST ist bei automatischem Neustart (Warmstart) wirkungslos.

Ungepufferter automatischer Neustart (Warmstart)

Wird Ihre CPU ohne Pufferbatterie betrieben (falls wartungsfreier Betrieb notwendig ist), wird nach dem Einschalten oder bei Spannungswiederkehr nach NETZ-AUS die CPU automatisch urgelöscht und anschließend ein Neustart (Warmstart) durchgeführt. Das Anwenderprogramm muß auf Flash-EPROM (Memory Card) vorhanden sein.

Wiederanlauf

Nach Spannungsausfall im RUN und anschließender Spannungswiederkehr durchlaufen S7-400-CPUs eine Initialisierungsroutine und führen dann automatisch einen Wiederanlauf durch. Bei einem Wiederanlauf wird das Anwenderprogramm an der Stelle fortgesetzt, an der die Bearbeitung unterbrochen wurde. Der Teil des Anwenderprogramms, der vor Netzausfall nicht mehr bearbeitet wurde, wird als Restzyklus bezeichnet. Der Restzyklus kann auch zeit- und alarmgesteuerte Programmteile enthalten.

Ein Wiederanlauf ist prinzipiell nur zulässig, wenn das Anwenderprogramm im STOP-Zustand nicht verändert wurde (z. B. durch Nachladen eines geänderten Bausteins) bzw. wenn nicht aus anderen Gründen Neustart (Warmstart) erforderlich ist. Man unterscheidet zwischen manuellem und automatischem Wiederanlauf.

Manueller Wiederanlauf

Ein manueller Wiederanlauf ist nur möglich bei entsprechender Parametrierung im Parametersatz der CPU und nach den folgenden STOP-Ursachen:

- der Betriebsartenschalter wurde von RUN auf STOP gesetzt
- vom Anwender programmierte STOPs, STOPs nach Aufruf nicht geladener OBs
- STOP-Zustand wurde vom PG oder durch Kommunikationsfunktion herbeigeführt.

Ein manueller Wiederanlauf kann ausgelöst werden:

- über den Betriebsartenschalter

  Der Schalter CRST/WRST muß auf WRST stehen.
- über Menübefehl vom PG aus bzw. über Kommunikationsfunktionen (wenn der Betriebsartenschalter auf RUN oder RUN-P steht).
- wenn im Parametersatz der CPU manueller Wiederanlauf parametriert wurde.

Automatischer Wiederanlauf

Ein automatischer Wiederanlauf kann ausgelöst werden bei NETZ-EIN, wenn

- die CPU bei NETZ-AUS nicht im STOP oder HALT war
- der Betriebsartenschalter auf RUN oder RUN-P steht
- im Parametersatz der CPU automatischer Wiederanlauf nach NETZ-EIN parametriert wurde.

Der Schalter CRST/WRST ist bei automatischem Wiederanlauf wirkungslos.

Remanente Datenbereiche nach Netzausfall

S7-300- und S7-400-CPUs reagieren unterschiedlich bei Spannungswiederkehr nach Netzausfall.

S7-300-CPUs (ausgenommen CPU 318) kennen nur die Anlaufart "Neustart". Mit STEP 7 können Sie jedoch die Remanenz von Merkern, Zeiten, Zählern und Bereichen in Datenbausteinen festlegen, um Datenverlust bei Spannungsausfall zu vermeiden. Bei Spannungswiederkehr erfolgt ein ”automatischer Neustart mit Gedächtnis”.

S7-400-CPUs reagieren auf Spannungswiederkehr abhängig von der Parametrierung entweder mit Neustart (Warmstart) (nach gepuffertem oder ungepuffertem NETZ-EIN) oder mit Wiederanlauf (nur nach gepuffertem NETZ-EIN möglich).

Die Tabelle A-1 zeigt das Remanenzverhalten der S7-300- und S7-400-CPUs bei Neustart (Warmstart), Kaltstart und Wiederanlauf:

X bedeutet	Daten bleiben erhalten
VC bedeutet	Codebaustein im EPROM bleibt erhalten, eventuell überladener Codebaustein geht verloren
VX bedeutet	Datenbaustein bleibt nur erhalten, wenn im EPROM vorhanden, remanente Daten werden aus NV-RAM übernommen (geladene oder erzeugte Datenbausteine im RAM gehen verloren)
0 bedeutet	Daten werden zurückgesetzt bzw. gelöscht (Inhalt von DBs)
V bedeutet	Daten werden auf den Vorbesetzungswert aus dem EPROM-Speicher gesetzt
--- bedeutet	nicht möglich, da kein NV-RAM vorhanden

Remanenzverhalten im Arbeitsspeicher (bei EPROM- und RAM-Ladespeicher):

EPROM (Memory Card oder Integriert)									
CPU mit Pufferung					**CPU ohne Pufferung**				
Daten	Bausteine im Lade-speicher	DB im Arbeits-speicher	Merker, Zeiten, Zähler	Merker, Zeiten, Zähler	Bausteine im Lade-speicher	DB im Arbeits-speicher	DB im Arbeits-speicher	Merker, Zeiten, Zähler	Merker, Zeiten, Zähler
			(rema-nent para-metriert)	(nicht rema-nent para-metriert)		(rema-nent parame-triert)	(nicht rema-nent para-metriert)	(rema-nent para-metriert)	(nicht rema-nent para-metriert)
Neustart bei S7-300	X	X	X	0	VC	VX	V	X	0
Neustart bei S7-400	X	X	X	0	VC	---	V	0	0
Kaltstart bei S7-300	X	X	0	0	VC	V	V	0	0
Kaltstart bei S7-400	X	X	0	0	VC	---	V	0	0
Wieder-anlauf bei S7-400	X	X	X	X		Nur	Neustart	erlaubt	

Anlauftätigkeiten

Welche Tätigkeiten die CPU beim Anlauf durchführt, zeigt nachfolgende Tabelle:

Tätigkeiten in der Bearbeitungsreihenfolge	**bei Neustart (Warmstart)**	**bei Kaltstart**	**bei Wiederanlauf**
U-Stack/B-Stack löschen	X	X	0
Nicht remanente Merker, Zeiten, Zähler löschen	X	0	0
Alle Merker, Zeiten, Zähler löschen	0	X	0
Prozeßabbild der Ausgänge löschen	X	X	parametrierbar
Ausgänge der Signalbaugruppen zurücksetzen	X	X	parametrierbar
Prozeßalarme verwerfen	X	X	0
Verzögerungsalarme verwerfen	X	X	0
Diagnosealarme verwerfen	X	X	X
Systemzustandsliste (SZL) aktualisieren	X	X	X

Tätigkeiten in der Bearbeitungsreihenfolge	bei Neustart (Warmstart)	bei Kaltstart	bei Wiederanlauf
Baugruppenparameter auswerten und an Baugruppen übergeben oder Defaultwerte übergeben	X	X	X
Bearbeitung des jeweiligen Anlauf-OB	X	X	X
Restzyklus (Teil des Anwenderprogramms, das aufgrund eines NETZ-AUS nicht weiterbearbeitet werden konnte) bearbeiten	0	0	X
Prozeßabbild der Eingänge aktualisieren	X	X	X
Digitale Ausgänge freigeben (Signal OD aufheben) nach Betriebszustandsübergang in RUN	X	X	X
X bedeutet wird durchgeführt 0 bedeutet wird nicht durchgeführt			

Abbrechen eines Anlaufs

Treten während des Anlaufs Fehler auf, wird der Anlauf abgebrochen, und die CPU geht oder bleibt in STOP.

Ein abgebrochener Neustart (Warmstart) muß wiederholt werden. Nach einem abgebrochenen Wiederanlauf ist sowohl ein Neustart (Warmstart) als auch ein Wiederanlauf möglich.

Ein Anlauf (Neustart/Wiederanlauf) wird nicht durchgeführt oder abgebrochen:

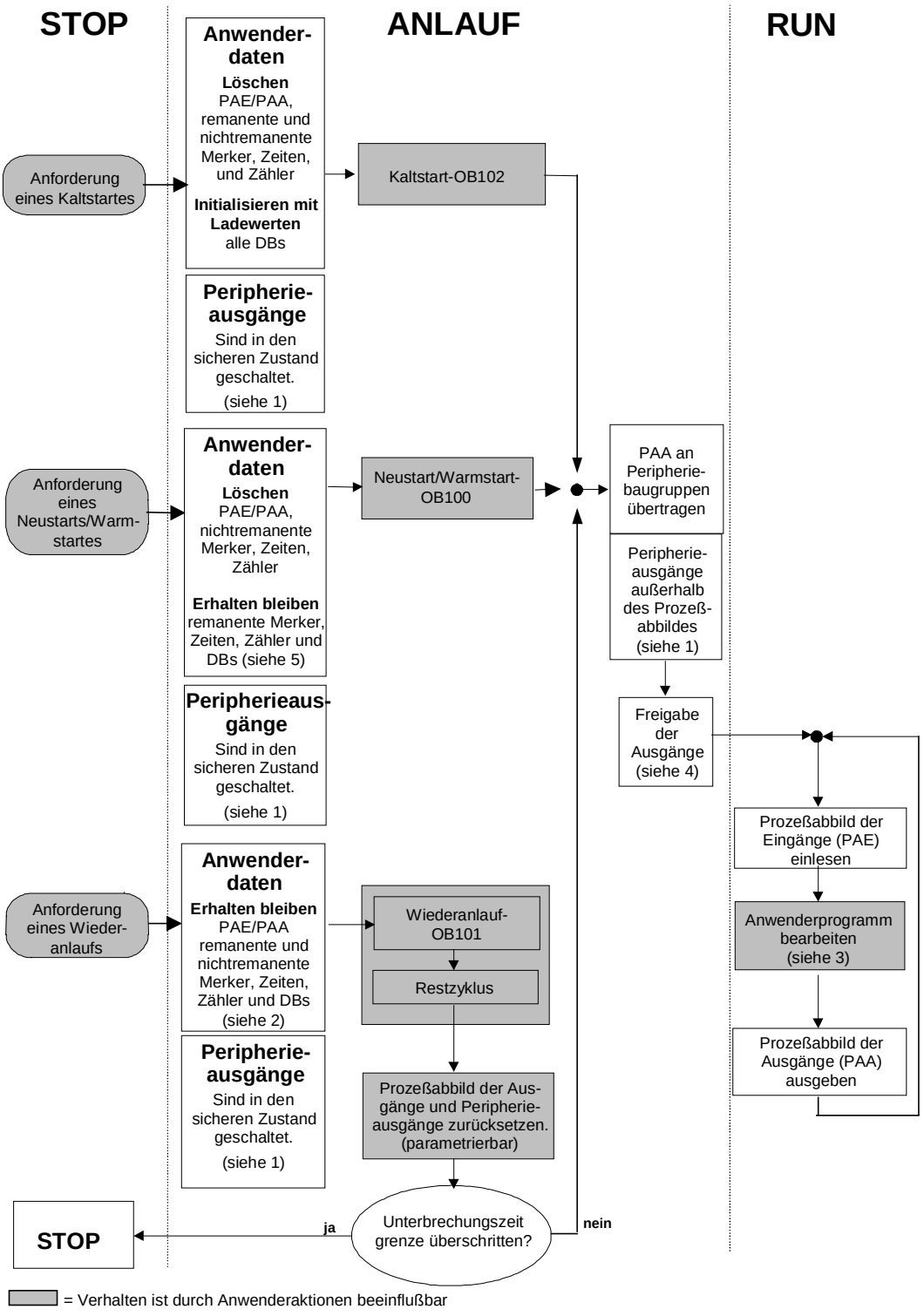
- wenn der Schlüsselschalter der CPU auf STOP steht
- Urlöschen angefordert wird
- eine Memory Card gesteckt ist, deren Anwendungskennung für STEP 7 nicht zulässig ist (z.B. STEP 5)
- im Einzelprozessorbetrieb mehr als eine CPU gesteckt ist
- wenn das Anwenderprogramm einen OB enthält, den die CPU nicht kennt oder der gesperrt wurde
- nach dem Einschalten der Netzspannung die CPU feststellt, daß nicht alle Baugruppen, die in der mit STEP 7 erstellten Konfigurationstabelle aufgeführt sind, tatsächlich gesteckt sind (Parametrierung Soll-Ist-Unterschied unzulässig)
- bei der Auswertung der Baugruppenparameter Fehler auftreten.

Ein Wiederanlauf wird auch nicht durchgeführt, oder er wird abgebrochen, wenn

- die CPU zuvor urgelöscht wurde (nach dem Urlöschen ist nur Neustart (Warmstart) zulässig)
- die Unterbrechungszeitgrenze überschritten ist (Die Unterbrechungszeit ist die Zeit, die nach dem Verlassen des RUN vergeht bis der Anlauf-OB inkl. Restzyklus bearbeitet wurde.)
- die Baugruppenkonfiguration geändert wurde (z.B. Baugruppentausch)
- durch Parametrierung nur Neustart (Warmstart) zugelassen ist
- wenn im STOP Bausteine geladen, gelöscht oder geändert wurden.

Ablauf

Nachfolgendes Bild zeigt die Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN.



Legende zum Bild "Tätigkeiten der CPU in den Betriebszuständen ANLAUF und RUN":

1. Alle Peripherieausgänge sind durch die Peripheriebaugruppen hardwareseitig in den sicheren Zustand geschaltet (Defaultwert ="0"). Dies ist unabhängig davon, ob sie im Anwenderprogramm innerhalb oder außerhalb des Prozeßabbildbereiches verwendet werden.

   Beim Einsatz von ersatzwertfähigen Signalbaugruppen kann das Verhalten der Ausgänge parametriert werden, z. B. letzter Wert halten.

2. Ist notwendig für die Bearbeitung des Restzyklus.
3. Auch den Alarm-OBs steht beim erstmaligem Aufruf ein aktuelles Prozeßabbild der Eingänge zur Verfügung.
4. Den Zustand der zentralen und dezentralen Peripherieausgänge im ersten Zyklus des Anwenderprogramms -können Sie durch folgende Maßnahmen festlegen:
   - Parametrierbare Ausgabebaugruppen verwenden, um Ersatzwerte ausgeben zu können oder um den letzen Wert zu halten.
   - Bei Wiederanlauf: CPU-Anlauf-Parameter "Ausgänge zurücksetzen bei Wiederanlauf" aktivieren, um eine "0" auszugeben (entspricht der Voreinstellung).
   - Ausgänge im Anlauf-OB (OB 100, OB 101, OB 102) vorbesetzen.
5. In ungepufferten S7-300-Systemen bleiben nur die als remanent projektierten DB-Bereiche erhalten.

A.1.4 Betriebszustand RUN

Im Betriebszustand "RUN" erfolgt die zyklische, zeit- und alarmgesteuerte Programmbearbeitung:

- das Prozeßabbild der Eingänge wird eingelesen
- das Anwenderprogramm wird abgearbeitet
- das Prozeßabbild der Ausgänge wird ausgegeben.

Der aktive Austausch von Daten zwischen CPUs über die Globaldaten-Kommunikation (Globaldatentabelle) und über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ist nur im RUN-Zustand möglich.

Nachfolgende Tabelle zeigt beispielhaft, wann Datenaustausch in verschiedenen Betriebszuständen möglich ist:

Art der Kommunikation	Betriebszustand der CPU 1	Richtung des Datenaustauschs	Betriebszustand der CPU 2
Globaldaten-Kommunikation	RUN	↔	RUN
	RUN	→	STOP/HALT
	STOP	←	RUN
	STOP	X	STOP
	HALT	X	STOP/HALT
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFBs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFBs	RUN	↔	RUN
Einseitige Kommunikation	RUN	→	RUN
über Kommunikations-SFCs	RUN	→	STOP/HALT
Zweiseitige Kommunikation über Kommunikations-SFCs	RUN	↔	RUN

↔ bedeutet Datenaustausch ist in beide Richtungen möglich
→ bedeutet Datenaustausch ist nur in eine Richtung möglich
X bedeutet Datenaustausch ist nicht möglich

A.1.5 Betriebszustand HALT

Der Betriebszustand HALT nimmt eine Sonderstellung ein. Er wird nur zu Testzwecken im Anlauf oder RUN eingenommen. Im Betriebszustand HALT:

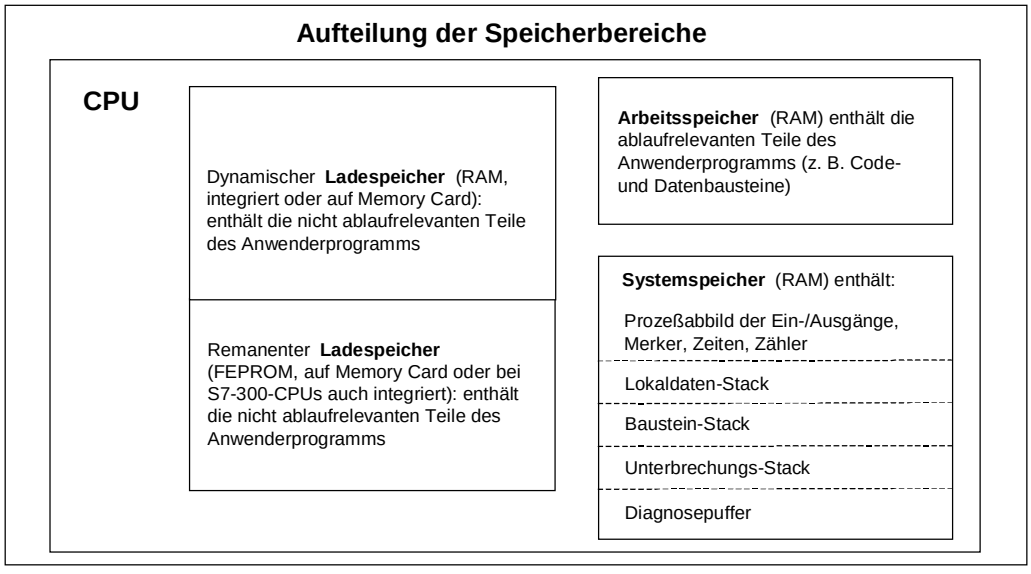
- werden alle Zeiten eingefroren: Zeiten und Betriebsstundenzähler werden nicht bearbeitet, Überwachungszeiten werden angehalten, die Grundtakte der zeitgesteuerten Ebenen werden angehalten.
- läuft die Echtzeituhr
- werden Ausgänge nicht freigeschaltet, können aber zu Testzwecken freigegeben werden
- können Ein- und Ausgänge gesteuert werden
- gehen gepufferte CPUs bei Netzausfall und -wiederkehr in HALT in STOP und führen keinen automatischen Wiederanlauf oder Neustart aus. Ungepufferte CPUs führen bei Netzwiederkehr einen ungepufferten automatischen Neustart aus.
- können auch Globaldaten empfangen und passiv einseitige Kommunikation über Kommunikations-SFBs für projektierte Verbindungen und über Kommunikations-SFCs für nichtprojektierte Verbindungen ausgeführt werden (siehe auch Tabelle in Betriebszustand RUN)

A.2 Speicherbereiche von S7-CPUs

A.2.1 Aufteilung der Speicherbereiche

Der Speicher der S7-CPUs läßt sich in drei Bereiche aufteilen (siehe nachfolgendes Bild):

- Der Ladespeicher dient zur Aufnahme des Anwenderprogramms ohne symbolische Operandenzuordnung oder Kommentare (diese bleiben im Speicher des PG). Der Ladespeicher kann RAM- oder EPROM-Speicher sein.
- Bausteine, die als nicht ablaufrelevant gekennzeichnet sind, werden ausschließlich in den Ladespeicher aufgenommen.
- Der Arbeitsspeicher (integrierter RAM) dient zur Aufnahme der für den Programmablauf relevanten Teile des S7-Programms. Die Programmbearbeitung erfolgt ausschließlich im Bereich von Arbeitsspeicher und Systemspeicher.
- Der Systemspeicher (RAM) enthält die Speicherelemente, die jede CPU dem Anwenderprogramm zur Verfügung stellt, wie z. B.: das Prozeßabbild der Ein- und Ausgänge, Merker, Zeiten und Zähler. Außerdem enthält der Systemspeicher den Baustein-Stack und den Unterbrechungs-Stack.
- Der Systemspeicher der CPU stellt außerdem temporären Speicher (Lokaldaten-Stack) zur Verfügung, der dem Programm beim Aufrufen eines Bausteins für dessen temporäre Daten zugeordnet wird. Diese Daten sind nur solange gültig, wie der Baustein aktiv ist.


Aufteilung der Speicherbereiche
CPU
Dynamischer Ladespeicher (RAM, integriert oder auf Memory Card): enthält die nicht ablaufrelevanten Teile des Anwenderprogramms
Remanenter Ladespeicher (FEPROM, auf Memory Card oder bei S7-300-CPUs auch integriert): enthält die nicht ablaufrelevanten Teile des Anwenderprogramms
Arbeitsspeicher (RAM) enthält die ablaufrelevanten Teile des Anwenderprogramms (z. B. Code- und Datenbausteine)
Systemspeicher (RAM) enthält:
Prozeßabbild der Ein-/Ausgänge, Merker, Zeiten, Zähler
Lokaldaten-Stack
Baustein-Stack
Unterbrechungs-Stack
Diagnosepuffer

A.2.2 Lade- und Arbeitsspeicher

Wenn Sie das Anwenderprogramm vom Programmiergerät in die CPU laden, werden nur die Code- und Datenbausteine in den Lade- und den Arbeitsspeicher der CPU geladen.

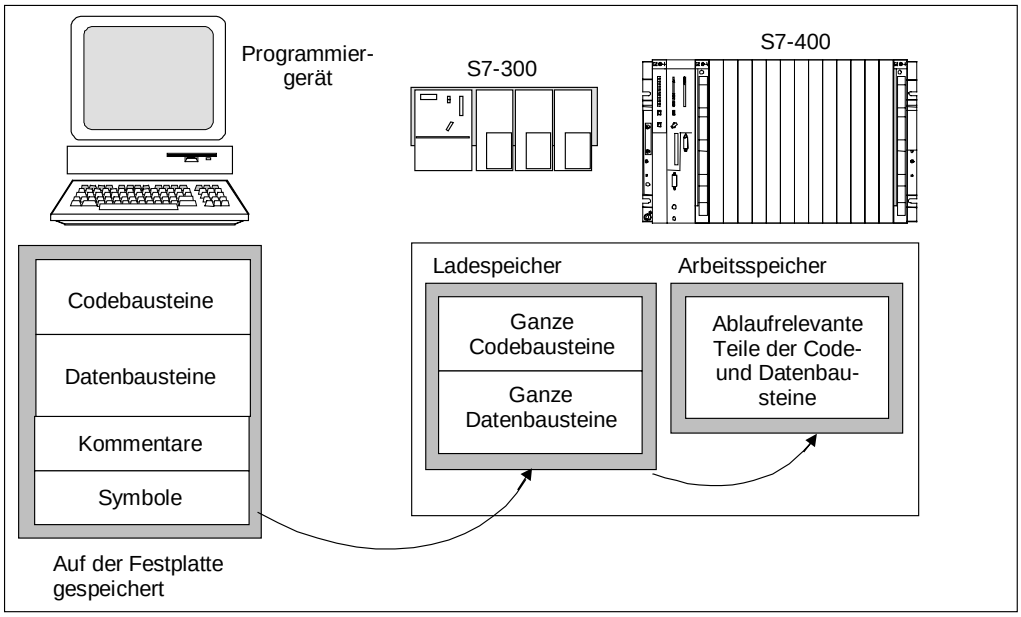
Die symbolische Operandenzuordnung (Symboltabelle) und die Bausteinkommentare bleiben im Speicherbereich des PG.

Aufteilung des Anwenderprogramms

Um eine schnelle Bearbeitung des Anwenderprogramms zu gewährleisten und den nicht erweiterbaren Arbeitsspeicher nicht unnötig zu belasten, werden nur die Teile der Bausteine, die für die Programmbearbeitung relevant sind, in den Arbeitsspeicher geladen.

Bausteinteile, die nicht erforderlich sind, um das Programm ablaufen zu lassen (z. B. Bausteinköpfe), bleiben im Ladespeicher.

Nachfolgendes Bild zeigt das Laden des Programms in den CPU-Speicher.



Hinweis

Datenbausteine, die mit Hilfe von Systemfunktionen (z. B. SFC 22 CREAT_DB) im Anwenderprogramm erstellt werden, speichert die CPU komplett in den Arbeitsspeicher.

Einige CPUs verfügen über getrennt verwaltete Bereiche für Code und Daten im Arbeitsspeicher. Größe und Belegung der Bereiche werden bei diesen CPUs auf der Registerseite "Speicher" des Baugruppenzustands angezeigt.

Kennzeichnen von Datenbausteinen als “nicht ablaufrelevant”

Datenbausteine, die als Teil eines AWL-Programms in einer Quelldatei programmiert werden, können als ”nicht ablaufrelevant” gekennzeichnet werden (Schlüsselwort UNLINKED). Das bedeutet, beim Laden in die CPU werden diese DBs nur im Ladespeicher abgelegt. Ihr Inhalt kann bei Bedarf mit Hilfe der SFC 20 BLKMOV in den Arbeitsspeicher kopiert werden.

Dadurch kann Platz im Arbeitsspeicher eingespart werden. Der erweiterbare Ladespeicher dient als Zwischenspeicher (z. B. für Rezepturen: nur die Rezeptur, die als nächste bearbeitet werden soll, wird in den Arbeitsspeicher geladen).

Struktur des Ladespeichers

Der Ladespeicher kann durch den Einsatz von Memory Cards erweitert werden. Die maximale Größe des Ladespeichers entnehmen Sie dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten".

Bei S7-300-CPUs kann der Ladespeicher außer einem integrierten RAM- auch einen integrierten EPROM-Anteil haben. Bereiche in Datenbausteinen können durch Parametrierung mit STEP 7 remanent erklärt werden (siehe Remanente Speicherbereiche in S7-300-CPUs).

Bei S7-400-CPUs ist der Einsatz einer Memory Card (RAM oder EPROM) zur Erweiterung des Ladespeichers unerläßlich. Der integrierte Ladespeicher ist ein RAM-Speicher und dient im wesentlichen dem Nachladen und Korrigieren von Bausteinen. Bei neuen S7-400-CPUs kann auch zusätzlicher Arbeitsspeicher gesteckt werden.

Verhalten des Ladespeichers bei RAM- und EPROM-Bereichen

Je nachdem, ob Sie eine RAM- oder EPROM-Memory Card zur Erweiterung des Ladespeichers wählen, kann sich beim Laden, Nachladen und Urlöschen ein unterschiedliches Verhalten des Ladespeichers ergeben.

Nachfolgende Tabelle zeigt die Lademöglichkeiten:

Speicherart	Lademöglichkeiten	Ladeart
RAM	Laden und Löschen von einzelnen Bausteinen	PG-CPU-Verbindung
	Laden und Löschen eines gesamten S7-Programmes	PG-CPU-Verbindung
	Nachladen von einzelnen Bausteinen	PG-CPU-Verbindung
EPROM integriert (nur bei S7-300) oder steckbar	Laden von gesamten S7-Programmen	PG-CPU-Verbindung
EPROM steckbar	Laden von gesamten S7-Programmen	Laden des EPROMs auf dem PG und Stecken der Memory Card in die CPU Laden des EPROMs auf der CPU

Programme, die im RAM gespeichert sind, gehen verloren, wenn Sie die CPU urlöschen (MRES) bzw. wenn Sie die CPU oder die RAM-Memory Card ziehen.

Programme, die auf EPROM-Memory-Cards gespeichert sind, gehen beim Urlöschen nicht verloren und bleiben auch ohne Batteriepufferung erhalten (Transport, Sicherungskopien).

A.2.3 Systemspeicher

A.2.3.1 Verwenden der Systemspeicherbereiche

Der Systemspeicher der S7-CPUs ist in Operandenbereiche aufgeteilt (siehe nachfolgende Tabelle). Durch Verwendung der entsprechenden Operationen adressieren Sie in Ihrem Programm die Daten direkt in den jeweiligen Operandenbereich.

Operanden-bereich	Zugriff über Einheiten der folgenden Größe:	S7-Notation	Beschreibung
Prozeßabbild der Eingänge	Eingang (Bit)	E	Zu Beginn jedes Zyklus liest die CPU die Eingänge aus den Eingabebaugruppen und speichert die Werte in das Prozeßabbild der Eingänge.
	Eingangsbyte	EB	
	Eingangswort	EW	
	Eingangsdoppelwort	ED	
Prozeßabbild der Ausgänge	Ausgang (Bit)	A	Während des Zyklus berechnet das Programm die Werte für die Ausgänge und legt sie im Prozeßabbild der Ausgänge ab. Am Ende des Zyklus schreibt die CPU die errechneten Ausgangswerte in die Ausgabebaugruppen.
	Ausgangsbyte	AB	
	Ausgangswort	AW	
	Ausgangsdoppelwort	AD	
Merker	Merker (Bit)	M	Dieser Bereich stellt Speicherplatz für im Programm errechnete Zwischenergebnisse zur Verfügung.
	Merkerbyte	MB	
	Merkerwort	MW	
	Merkerdoppelwort	MD	
Zeiten	Zeit (T)	T	In diesem Bereich stehen Zeiten zur Verfügung.
Zähler	Zähler (Z)	Z	In diesem Bereich stehen Zähler zur Verfügung.

Operanden-bereich	**Zugriff über Einheiten der folgenden Größe:**	**S7-Notation**	**Beschreibung**
Daten-baustein	Datenbaustein, geöffnet mit ”AUF DB”:	DB	Datenbausteine speichern Informationen für das Programm. Sie können entweder so definiert sein, daß alle Codebausteine auf sie zugreifen können (globale DBs), oder sie sind einem bestimmten FB oder SFB zugeordnet (Instanz-DB).
	Datenbit	DBX	
	Datenbyte	DBB	
	Datenwort	DBD	
	Datendoppelwort	DBW	
	Datenbaustein, geöffnet mit ”AUF DI”:	DI	
	Datenbit	DIX	
	Datenbyte	DIB	
	Datenwort	DIW	
	Datendoppelwort	DID	
Lokaldaten	Lokaldatenbit	L	Dieser Speicherbereich nimmt die temporären Daten eines Bausteins für die Dauer der Bearbeitung dieses Bausteins auf. Der L-Stack stellt auch Speicher zum Übertragen von Bausteinparametern und zum Speichern von Zwischenergebnissen aus KOP-Netzwerken zur Verfügung.
	Lokaldatenbyte	LB	
	Lokaldatenwort	LW	
	Lokaldatendoppelwort	LD	
Peripherie-bereich: Eingänge	Peripherieeingangsbyte	PEB	Die Peripheriebereiche der Ein- und Ausgänge erlauben direkten Zugriff auf zentrale und dezentrale Ein- und Ausgabebaugruppen.
	Peripherieeingangswort	PEW	
	Peripherieeingangs-doppelwort	PED	
Peripherie-bereich: Ausgänge	Peripherieausgangsbyte	PAB	
	Peripherieausgangswort	PAW	
	Peripherieausgangs-doppelwort	PAD	

Welche Adreßbereiche bei Ihrer CPU möglich sind, entnehmen Sie folgenden CPU-Beschreibungen bzw. Operationslisten **:**

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten".
- Referenzhandbuch "Automatisierungssysteme S7-400/M7-400. Baugruppendaten".
- Operationsliste "Automatisierungssystem S7-300".
- Operationsliste "Automatisierungssystem S7-400".

A.2.3.2 Prozeßabbild der Ein-/Ausgänge

Werden im Anwenderprogramm die Operandenbereiche Eingänge (E) und Ausgänge (A) angesprochen, werden nicht die Signalzustände auf den digitalen Signalbaugruppen abgefragt, sondern es wird auf einen Speicherbereich im Systemspeicher der CPU und der dezentralen Peripherie zugegriffen. Diesen Speicherbereich bezeichnet man als Prozeßabbild.

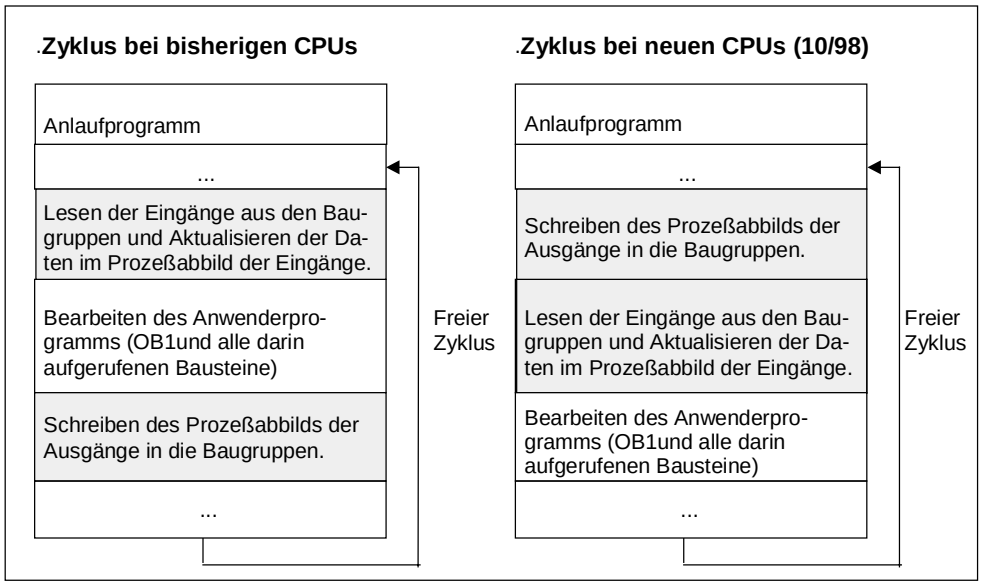
Das Prozeßabbild ist in zwei Teile gegliedert: das Prozeßabbild der Eingänge und das Prozeßabbild der Ausgänge.

Voraussetzung für den Zugriff auf das Prozeßabbild

Die CPU kann nur auf das Prozeßabbild der Baugruppen zugreifen, die Sie mit STEP 7 konfiguriert haben.

Aktualisieren des Prozeßabbilds

Das Prozeßabbild wird vom Betriebssystem zyklisch aktualisiert. Nachfolgendes Bild zeigt die Bearbeitungsschritte innerhalb eines Zyklus, unterschieden nach den bisherigen CPUs und nach den neuen ab 10/98 erhältlichen CPUs.


Zyklus bei bisherigen CPUs
Anlaufprogramm
...
Lesen der Eingänge aus den Baugruppen und Aktualisieren der Daten im Prozeßabbild der Eingänge.
Bearbeiten des Anwenderprogramms (OB1und alle darin aufgerufenen Bausteine)
Schreiben des Prozeßabbilds der Ausgänge in die Baugruppen.
...
Freier Zyklus
Zyklus bei neuen CPUs (10/98)
Anlaufprogramm
...
Schreiben des Prozeßabbilds der Ausgänge in die Baugruppen.
Lesen der Eingänge aus den Baugruppen und Aktualisieren der Daten im Prozeßabbild der Eingänge.
Bearbeiten des Anwenderprogramms (OB1und alle darin aufgerufenen Bausteine)
...
Freier Zyklus

Vorteile des Prozeßabbildes

Der Zugriff auf das Prozeßabbild hat gegenüber dem direkten Zugriff auf die Ein-/Ausgabebaugruppen den Vorteil, daß der CPU für die Dauer der zyklischen Programmbearbeitung ein konsistentes Abbild der Prozeßsignale zur Verfügung steht. Wenn sich während der Programmbearbeitung ein Signalzustand auf einer Eingabebaugruppe ändert, bleibt der Signalzustand im Prozeßabbild erhalten bis zur Prozeßabbildaktualisierung im nächsten Zyklus. Außerdem benötigt der Zugriff auf das Prozeßabbild wesentlich weniger Zeit als der direkte Zugriff auf die Signalbaugruppen, weil sich das Prozeßabbild im internen Speicher der CPU befindet.

Teilprozeßabbilder

Neben dem (Gesamt-) Prozeßabbild (PAE und PAA) können Sie bei S7-400 für eine CPU bis zu 15 Teil-prozeßabbilder parametrieren (CPU-spezifisch, Nr. 1 bis max. Nr. 15; siehe "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und Referenzhandbuch "Automatisierungssysteme S7-400/M7-400, Baugruppendaten"). Damit haben Sie die Möglichkeit, unabhängig von der zyklischen Aktualisierung des Prozeßabbilds, Teilbereiche des Prozeßabbildes bei Bedarf zu aktualisieren.

Jede Ein-/Ausgangsadresse, die Sie mit STEP 7 einem Teilprozeßabbild zugewiesen haben, gehört nicht mehr zum OB 1-Prozeßabbild der Ein-/Ausgänge!

Das Teilprozeßabbild definieren Sie mit STEP 7 bei der Adreßzuweisung (welche Ein-/Ausgangsadressen der Baugruppen in welchem Teilprozeßabbild geführt werden). Die Aktualisierung des Teilprozeßabbilds erfolgt entweder mit Hilfe von SFCs oder systemseitig durch Kopplung an einen OB.

Hinweis

Bei S7-300 CPUs können nicht belegte Prozeßabbildein- und -ausgänge als zusätzliche Merkerbereiche verwendet werden. Programme, die diese Möglichkeit nutzen, sind nur unter folgender Bedingung auf älteren (d.h. vor 4/99) S7-400-CPUs ablauffähig:

Bei diesen S7-400 CPUs

- müssen die als Merker verwendeten Prozeßabbilder außerhalb der parametrierten "Größe des Prozeßabbildes" liegen bzw.
- müssen in einem Teilprozeßabbild liegen, das weder systemseitig noch durch SFC 26/SFC 27 aktualisiert wird!

Teilprozeßabbilder aktualisieren mit SFCs

Sie können mit Hilfe von SFCs das gesamte Prozeßabbild oder Teilprozeßabbilder vom Anwenderprogramm aus aktualisieren.

Voraussetzung: Das entsprechende Teilprozeßabbild wird nicht systemseitig aktualisiert!

- SFC 26 UPDAT_PI: Prozeßabbild der Eingänge aktualisieren
- SFC 27 UPDAT_PO: Prozeßabbild der Ausgänge aktualisieren.

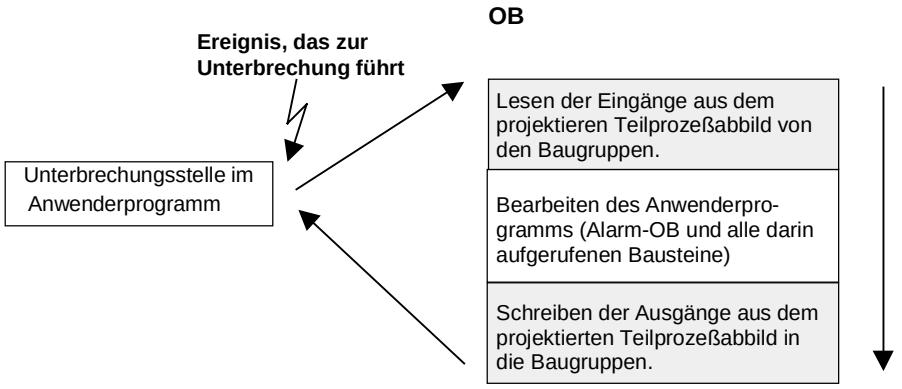
Teilprozeßabbilder systemseitig aktualisieren

Sie können Teilprozeßabbilder auch automatisch vom Betriebssystem bei Aufruf eines OB aktualisieren lassen – ähnlich dem (Gesamt-) Prozeßabbild, das zyklisch vor bzw. nach der OB 1-Bearbeitung aktualisiert wird. Diese Funktion ist nur für bestimmte CPUs parametrierbar.

Im Betrieb wird dann automatisch das zugeordnete Teilprozeßabbild aktualisiert:

- vor der OB-Bearbeitung das Teilprozeßabbild der Eingänge
- nach der OB-Bearbeitung das Teilprozeßabbild der Ausgänge

Welches Teilprozeßabbild welchem OB zugeordnet ist, parametrieren Sie für die CPU zusammen mit der Priorität des OBs.


OB
Ereignis, das zur Unterbrechung führt
Unterbrechungsstelle im Anwenderprogramm
Lesen der Eingänge aus dem projektieren Teilprozeßabbild von den Baugruppen.
Bearbeiten des Anwenderpro-gramms (Alarm-OB und alle darin aufgerufenen Bausteine)
Schreiben der Ausgänge aus dem projektierten Teilprozeßabbild in die Baugruppen.

Peripheriezugriffsfehler (PZF) bei Prozeßabbild-Aktualisierung

Die voreingestellte Reaktion der CPU-Familien (S7-300 und S7-400) ist bei einem Fehler während der Prozeßabbild-Aktualisierung unterschiedlich:

- S7-300: Kein Diagnosepuffer-Eintrag, kein OB-Aufruf; die entsprechenden Eingangs-/Ausgangsbytes werden auf 0 gesetzt.
- S7-400: Diagnosepuffer-Eintrag und Start des OB 85 bei jedem Zugriff auf die Peripherie bei jeder Aktualisierung des entsprechenden Prozeßabbildes. Die fehlerhaften Eingangs-/Ausgangsbytes werden auf 0 gesetzt.

Bei neuen CPUs (ab 4/99) können Sie die Reaktion bei Peripheriezugriffsfehlern umparametrieren, so daß die CPU

- nur bei kommendem und gehendem PZF einen Diagnosepuffer-Eintrag erzeugt und den OB 85 startet oder
- das voreingestellte Verhalten der S7-300 zeigt (kein OB-85-Aufruf) oder
- das voreingestellte Verhalten der S7-400 zeigt (OB-85-Aufruf bei jedem einzelnen Zugriff)

Wie oft startet der OB 85?

Neben der parametrierten Reaktion auf PZF (kommend/gehend oder bei jedem Peripheriezugriff) hat auch der Adreßraum einer Baugruppe Einfluß auf die Häufigkeit des OB-85-Starts:

Bei einer Baugruppe mit einem Adreßraum bis zu einem Doppelwort startet der OB 85 einmal, z.B. bei einer Digitalbaugruppe mit bis zu 32 Eingängen oder Ausgängen, oder einer Analogbaugruppe mit 2 Kanälen.

Bei Baugruppen mit größerem Adreßraum startet der OB 85 so oft, wie mit Doppelwortbefehlen darauf zugegriffen werden muß, z.B. bei einer 4kanaligen Analogbaugruppe zweimal.

A.2.3.3 Lokaldaten-Stack

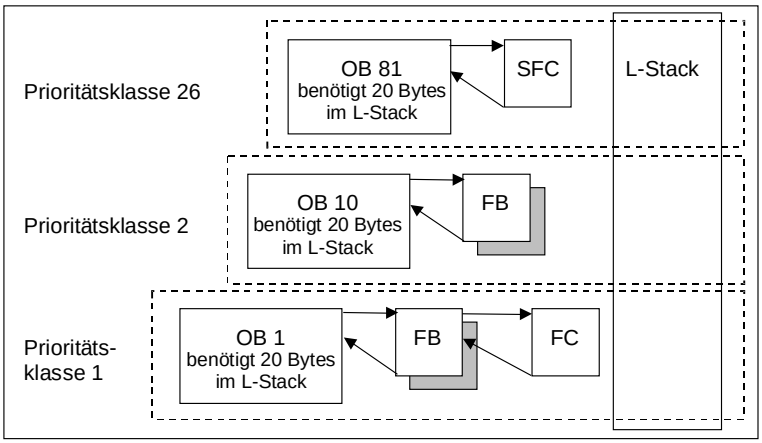
Der L-Stack speichert:

- die temporären Variablen der Lokaldaten von Bausteinen
- die Startinformation der Organisationsbausteine
- Informationen zum Übergeben von Parametern
- Zwischenergebnisse der Logik in Kontaktplan-Programmen

Beim Erstellen von Organisationsbausteinen können Sie temporäre Variablen (TEMP) deklarieren, die nur während der Bearbeitung des Bausteins zur Verfügung stehen und dann wieder überschrieben werden. Vor dem ersten Zugriff müssen die Lokaldaten initialisiert werden. Außerdem benötigt jeder Organisationsbaustein für seine Startinformation 20 Lokaldaten-Byte.

Die CPU besitzt einen begrenzten Speicher für die temporären Variablen (Lokaldaten) gerade bearbeiteter Bausteine. Die Größe dieses Speicherbereichs, des Lokaldaten-Stacks, ist CPU-abhängig. Der Lokaldaten-Stack wird zu gleichen Teilen unter den Prioritätsklassen aufgeteilt (Voreinstellung). Das bedeutet, jede Prioritätsklasse verfügt über einen eigenen Lokaldatenbereich. Damit ist gewährleistet, daß auch hochpriore Prioritätsklassen und ihre zugeordneten OBs Platz für ihre Lokaldaten zur Verfügung haben.

Nachfolgendes Bild zeigt die Zuordnung von Lokaldaten zu den Prioritätsklassen in einem Beispiel, in dem im L-Stack der OB 1 vom OB 10 unterbrochen wird, der dann wiederum vom OB 81 unterbrochen wird.



Vorsicht

Alle temporäre Variablen (TEMP) eines OB und seiner unterlagerten Bausteine werden im L-Stack gespeichert. Wenn Sie viele Schachtelungsebenen in Ihrer Bausteinbearbeitung verwenden, kann der L-Stack überlaufen.

S7-CPUs wechseln in den Betriebszustand "STOP", wenn Sie die zulässige L-Stack-Größe für ein Programm überschreiten.

Testen Sie den L-Stack (die temporären Variablen) in Ihrem Programm.

Berücksichtigen Sie den Lokaldatenbedarf von Synchronfehler-OBs.

Zuordnen von Lokaldaten zu Prioritätsklassen

Nicht jede Prioritätsklasse benötigt gleichviel Speicher im Lokaldaten-Stack. Durch Parametrierung mit STEP 7 können Sie bei S7-400-CPUs und bei der CPU 318 die Größe des Lokaldatenbereichs für die einzelnen Prioritätsklassen unterschiedlich festlegen. Nicht benötigte Prioritätsklassen können Sie abwählen. Bei S7-400-CPUs und der CPU 318 wird damit der Speicherbereich für andere Prioritätsklassen erweitert. Abgewählte OBs werden bei der Programmbearbeitung nicht berücksichtigt, dadurch sparen Sie Rechenzeit.

Bei den anderen S7-300-CPUs ist jeder Prioritätsklasse eine feste Anzahl von Lokaldaten zugeordnet (256 Bytes), die nicht verändert werden kann.

A.2.3.4 Unterbrechungs-Stack

Wird die Programmbearbeitung durch einen OB mit höherer Priorität unterbrochen, speichert das Betriebssystem die aktuellen Inhalte der Akkumulatoren und Adreßregister sowie die Nummer und die Größe der geöffneten Datenbausteine im U-Stack.

Ist die Bearbeitung des neuen OB beendet, lädt das Betriebssystem die Informationen aus dem U-Stack und nimmt die Bearbeitung des unterbrochenen Bausteins an der Stelle wieder auf, an der die Unterbrechung eintrat.

Im Betriebszustand "STOP" können Sie mit STEP 7 am PG den U-Stack auslesen. Damit können Sie leichter die Ursache dafür erkennen, warum die CPU in den Betriebszustand "STOP" gewechselt hat.

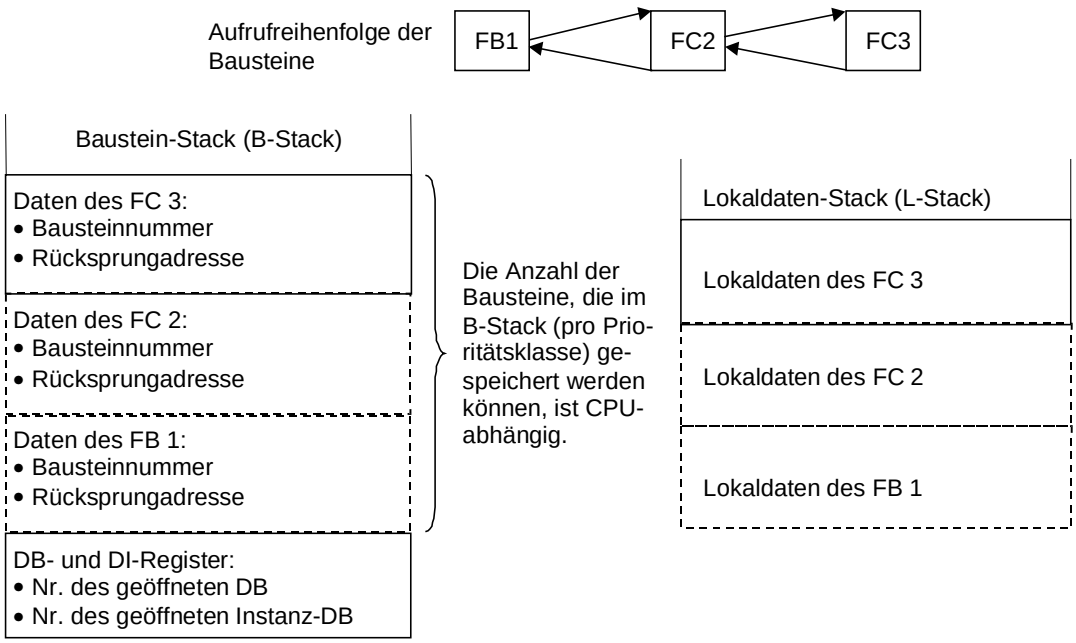
A.2.3.5 Baustein-Stack

Wird die Bearbeitung eines Bausteins durch den Aufruf eines anderen Bausteins oder durch eine höhere Prioritätsklasse (Alarm/Fehlerbehandlung) unterbrochen, speichert der B-Stack folgende Daten:

- Nummer, Art (OB, FB, FC, SFB, SFC) und Rücksprungadresse des Bausteins, der unterbrochen wurde.
- Nummer der Datenbausteine (aus DB- und DI-Register), die zum Zeitpunkt der Unterbrechung geöffnet waren.

Mit den gespeicherten Daten kann das Anwenderprogramm nach der Unterbrechung wieder fortgesetzt werden.

Befindet sich die CPU im Betriebszustand "STOP", können Sie den B-Stack mit STEP 7 am PG anzeigen lassen. Der B-Stack führt alle Bausteine auf, deren Bearbeitung zu dem Zeitpunkt, als die CPU in den Betriebszustand "STOP" versetzt wurde, nicht beendet war. Die Bausteine werden in der Reihenfolge aufgelistet, in der die Bearbeitung begonnen wurde (siehe nachfolgendes Bild).



Datenbausteinregister

Es existieren zwei Datenbausteinregister. Sie beinhalten die Nummern der aufgeschlagenen Datenbausteine

- im DB-Register steht die Nummer des aufgeschlagenen globalen Datenbausteins
- im DI-Register steht die Nummer des aufgeschlagenen Instanz-Datenbausteins.

A.2.3.6 Auswertung des Diagnosepuffers

Ein Teil der Systemzustandsliste ist der Diagnosepuffer, in den zu System-Diagnoseereignissen und anwenderdefinierten Diagnoseereignissen in der Reihenfolge ihres Auftretens nähere Informationen eingetragen werden. Die Information, die beim Auftreten eines System-Diagnoseereignisses in den Diagnosepuffer eingetragen wird, ist identisch mit der Startinformation, die an den entsprechenden Organisationsbaustein übergeben wird.

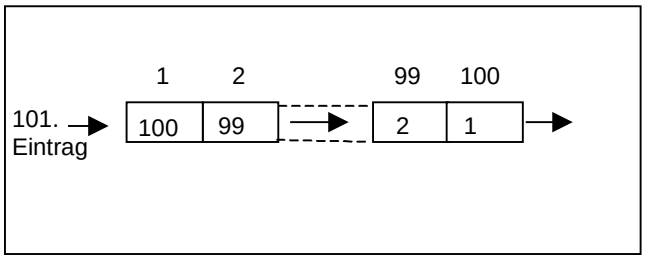
Die Einträge im Diagnosepuffer können nicht gelöscht werden, der Inhalt des Diagnosepuffers bleibt auch nach dem Urlöschen erhalten.

Der Diagnosepuffer bietet die Möglichkeit:

- bei Anlagenstopp die letzten Ereignisse vor dem STOP auszuwerten und die STOP-Ursache zu finden
- Fehlerursachen schneller zu erkennen und dadurch die Verfügbarkeit der Anlage zu erhöhen
- das dynamische Anlagenverhalten auszuwerten und zu optimieren.

Organisation des Diagnosepuffers

Der Diagnosepuffer ist als Ringpuffer für eine baugruppen-abhängige maximale Anzahl von Einträgen ausgelegt. Wird die maximale Anzahl von Einträgen erreicht, so wird bei einem neuen Diagnosepuffererereignis der älteste Eintrag gelöscht. Alle Einträge rücken entsprechend weiter. Dadurch steht der jüngste Diagnoseeintrag immer an erster Stelle. Für die S7-300 CPU 314 sind dies z. B. 100 Einträge :



Die Anzahl der angezeigten Einträge im Diagnosepuffer ist abhängig von der Baugruppe und ihrem aktuellen Betriebszustand. Bei bestimmten CPUs ist die Länge des Diagnosepuffers parametrierbar.

Inhalt des Diagnosepuffers

Das **obere** Listenfeld enthält eine Liste aller aufgetretenen Diagnoseereignisse mit folgenden Informationen:

- Laufende Nummer des Eintrags (das neueste Ereignis hat die Nr. 1)
- Uhrzeit und Datum des Diagnoseereignisses: Angezeigt werden Uhrzeit und Datum der Baugruppe, falls auf der Baugruppe eine Uhr vorhanden ist. Um diese Zeitangaben sinnvoll nutzen zu können, ist es wichtig, daß Sie die Uhrzeit und das Datum auf der Baugruppe einstellen und gelegentlich überprüfen.
- Ereignistext (Kurzdarstellung)

Im **unteren** Textfeld werden zusätzliche Informationen zu dem im oberen Fenster ausgewählten Ereignis angezeigt. Dazu gehören z. B.:

- Ereignisnummer,
- Bezeichnung des Ereignisses,
- Betriebszustandswechsel, der durch das Diagnoseereignis verursacht wurde,
- Verweis auf Fehlerstelle in einem Baustein (Bausteintyp, -nummer, Relativadresse), die zum Eintrag des Ereignisses führte,
- Kommendes oder gehendes Ereignis.
- Ereignisspezifische Zusatzinformationen

Mit der Schaltfläche ”Hilfe zum Ereignis” können Sie sich zusätzliche Information zu dem im Listenfeld markierten Ereignis anzeigen lassen.

Erläuterungen zu den Ereignis-IDs finden Sie in der Referenzhilfe zu Systembausteinen und Systemfunktionen (Sprünge in Sprachbeschreibungen, Baustein-Hilfen, Systemattribute).

Speichern des Inhalts in einer Textdatei

Mit der Schaltfläche ”Speichern unter” im Register "Diagnosepuffer" des Dialogfelds "Baugruppenzustand" können Sie den Inhalt des Diagnosepuffers als ASCII-Text speichern.

Diagnosepuffer auslesen

Sie können sich den Inhalt des Diagnosepuffers über das Dialogfeld "Baugruppenzustand", Register "Diagnosepuffer" am PG/PC anzeigen lassen oder in einem Programm über die SFC 51 RDSYSST auslesen.

Letzter Eintrag vor STOP

Sie können festlegen, daß der letzte Diagnosepuffereintrag vor dem Übergang von RUN in STOP automatisch an ein angemeldetes Beobachtungsgerät (z. B. PG, OP, TD) gesendet wird, um sicherzustellen, daß die Ursache für den Betriebszustandswechsel in STOP schneller gefunden und behoben wird.

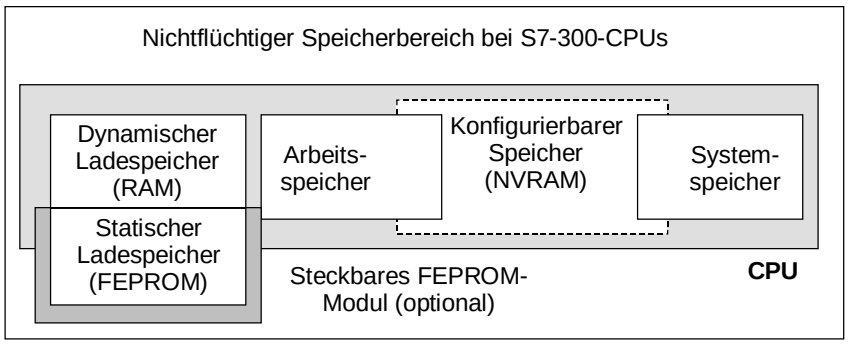
A.2.3.7 Remanente Speicherbereiche in S7-300-CPUs

Bei Stromausfall oder Urlöschen (MRES) wird der Speicher der S7-300-CPU (dynamischer Ladespeicher (RAM), Arbeitsspeicher und Systemspeicher) zurückgesetzt, wobei alle Daten, die in diesen Bereichen gespeichert sind, verloren gehen. S7-300-CPUs bieten die folgenden Möglichkeiten, Ihr Programm und seine Daten zu sichern:

- Sie können alle Daten, die sich im Ladespeicher, Arbeitsspeicher und in Teilen des Systemspeichers befinden, durch eine Batterie puffern.
- Sie können Ihr Programm im EPROM speichern (entweder Memory Card oder in die CPU integriert, siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten").
- Sie können eine CPU-abhängige Menge Daten in einem Bereich des nichtflüchtigen NVRAM sichern.

Verwenden des NVRAM

Ihre S7-300-CPU stellt einen Bereich im NVRAM (nichtflüchtiger RAM) zur Verfügung (siehe nachfolgendes Bild). Wenn Sie Ihr Programm im EPROM des Ladespeichers abgelegt haben, können Sie einige Daten (bei Stromausfall oder beim Übergang der CPU von STOP in RUN) auch durch eine entsprechende Konfiguration speichern.



Hierzu stellen Sie Ihre CPU so ein, daß die folgenden Daten in dem nichtflüchtigen RAM gespeichert werden:

- Daten, die in einem DB gespeichert sind (dies ist nur nützlich, wenn Sie auch Ihr Programm in einem EPROM des Ladespeichers abgelegt haben)
- Werte von Zeiten und Zählern
- Daten, die in Merkern gespeichert sind

Bei jeder CPU können Sie eine bestimmte Anzahl an Zeiten, Zählern und Merkern puffern. Außerdem wird eine spezifische Anzahl an Bytes zur Verfügung gestellt, in denen die Daten, die in DBs abgelegt sind, gespeichert werden können.

Die MPI-Adresse Ihrer CPU ist im NVRAM abgelegt. Damit ist sichergestellt, daß Ihre CPU nach einem Stromausfall oder Urlöschen noch kommunikationsfähig ist.

Verwenden der Batteriepufferung zum Sichern von Daten

Durch Batteriepufferung werden der Ladespeicher und der Arbeitsspeicher bei Stromausfall remanent. Wenn durch Ihre Konfiguration Zeiten, Zähler und Merker im NVRAM gespeichert werden, gehen unabhängig von der Batteriepufferung auch diese Informationen nicht verloren.

Konfigurieren der Daten des NVRAM

Wenn Sie die CPU mit STEP 7 konfigurieren, können Sie festlegen, welche Speicherbereiche remanent sein sollen.

Wieviel Speicher im NVRAM konfiguriert werden kann, ist CPU-abhängig. Sie können nicht mehr Daten puffern, als die für Ihre CPU angegebene Menge.

A.2.3.8 Remanente Speicherbereiche in S7-400-CPUs

Ungepufferter Betrieb

Bei Stromausfall oder Urlöschen (MRES) im ungepufferten Betrieb wird der Speicher der S7-400-CPU (dynamischer Ladespeicher (RAM), Arbeitsspeicher und Systemspeicher) zurückgesetzt, wobei alle Daten, die in diesen Bereichen gespeichert sind, verloren gehen.

Im ungepufferten Betrieb ist nur Neustart möglich und es gibt keine remanenten Speicherbereiche. Nur die MPI-Parameter (z. B. MPI-Adresse der CPU) bleiben nach Spannungsausfall erhalten. Damit ist die CPU nach Stromausfall oder Urlöschen noch kommunikationsfähig.

Gepufferter Betrieb

Im gepufferten Betrieb bleibt

- beim Wiederanlauf nach Spannungsausfall der Inhalt aller RAM-Bereiche vollständig erhalten.
- beim Neustart werden die Operandenbereiche Merker, Zeiten, Zähler gelöscht. Die Inhalte von Datenbausteinen bleiben erhalten.
- der Inhalt des RAM-Arbeitsspeichers bis auf nicht remanent parametrierte Merker, Zeiten, Zähler erhalten.

Konfigurieren remanenter Datenbereiche

Sie können eine CPU-abhängige Menge von Merkern, Zeiten, Zählern remanent erklären. Beim Neustart im gepufferten Betrieb bleiben diese Daten dann erhalten.

Durch Parametrierung mit STEP 7 legen Sie fest, welche Merker, Zeiten, Zähler bei einem Neustart remanent sein sollen. Sie können nicht mehr Daten puffern, als die für Ihre CPU zulässige Menge.

Nähere Informationen zur Parametrierung remanenter Speicherbereiche entnehmen Sie dem Referenzhandbuch "Automatisierungssysteme S7-400/M7-400. Baugruppendaten".

A.2.3.9 Konfigurierbare Speicherobjekte im Arbeitsspeicher

Bei einigen CPUs kann die Größe von Objekten wie Lokaldaten oder Diagnosepuffer unter HW Konfig eingestellt werden. Wenn Sie beispielsweise die voreingestellten Werte verkleinern, steht ein größerer Anteil des Arbeitsspeichers anderweitig zur Verfügung. Die Einstellungen dieser CPUs sind im Register "Speicher" des Baugruppenzustands (Schaltfläche Details) ablesbar.

Nach Ändern der Speicherkonfiguraton und Laden auf das Zielsystem ist ein Kaltstart des Zielsystems erforderlich, damit die Änderungen wirksam werden.

A.3 Daten- und Parametertypen

A.3.1 Einführung zu Daten- und Parametertypen

Alle in einem Anwenderprogramm verwendeten Daten müssen durch einen Datentyp gekennzeichnet sein. Man unterscheidet zwischen:

- elementaren Datentypen, die Ihnen STEP 7 zur Verfügung stellt
- zusammengesetzten Datentypen, die Sie erzeugen können, indem Sie elementare Datentypen verknüpfen und
- Parametertypen, mit denen Sie Parameter definieren, die an FBs oder FCs übergeben werden sollen.

Allgemeine Information

AWL-, FUP- oder KOP-Operationen arbeiten mit Datenobjekten bestimmter Größen. Bitverknüpfungsoperationen arbeiten beispielsweise mit Bits. Lade- und Transferoperationen (AWL) sowie Übertragungsoperationen (FUP und KOP) arbeiten mit Bytes, Wörtern und Doppelwörtern.

Bit nennt man eine Binärziffer ”0” oder ”1”. Ein Byte besteht aus 8 Bits, ein Wort aus 16 Bits und ein Doppelwort aus 32 Bits.

Arithmetische Operationen arbeiten ebenfalls mit Bytes, Wörtern oder Doppelwörtern. In diesen Byte-, Wort- oder Doppelwortoperanden können Sie Zahlen verschiedener Formate codieren, wie z. B. Ganzzahlen und Gleitpunktzahlen.

Wenn Sie die symbolische Adressierung verwenden, dann definieren Sie Symbole und geben einen Datentyp für diese Symbole an (siehe nachfolgende Tabelle). Verschiedene Datentypen haben verschiedene Formatoptionen und Zahlendarstellungen.

Dieses Kapitel beschreibt nur einige der Schreibweisen, die für Zahlen und Konstanten möglich sind. Nachfolgende Tabelle listet Formate von Zahlen und Konstanten, auf die nicht ausführlich eingegangen wird.

Format	Größe in Bits	Zahlendarstellung
Hexadezimal	8, 16 und 32	B#16#, W#16# und DW#16#
Binär	8, 16 und 32	2#
IEC-Datum	16	D#
IEC-Zeit	32	T#
Uhrzeit	32	TOD#
Zeichen	8	'A'

A.3.2 Elementare Datentypen

Jeder elementare Datentyp hat eine definierte Länge. Nachfolgende Tabelle listet die elementaren Datentypen auf.

Typ und Beschreibung	Größe in Bits	Formatoptionen	Bereich und Zahlendarstellung (niedrigster bis höchster Wert)	Beispiel
BOOL(Bit)	1	Bool-Text	TRUE/FALSE	TRUE
BYTE(Byte)	8	Hexadezimalzahl	B16#0 bis B16#FF	L B#16#10 L byte#16#10
WORD (Wort)	16	Dualzahl Hexadezimalzahl BCD Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111 W#16#0 bis W#16#FFFF C#0 bis C#999 B#(0,0) bis B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (Doppelwort)	32	Dualzahl Hexadezimalzahl Dezimalzahl ohne Vorzeichen	2#0 bis 2#1111_1111_1111_1111_ 1111_1111_1111_1111 DW#16#0000_0000 bis DW#16#FFFF_FFFF B#(0,0,0,0) bis B#(255,255,255,255)	2#1000_0001_0001_1000 _ 1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (Ganzzahl)	16	Dezimalzahl mit Vorzeichen	-32768 bis 32767	L 1
DINT (Ganzzahl, 32 Bit)	32	Dezimalzahl mit Vorzeichen	L#-2147483648 bis L#2147483647	L L#1
REAL (Gleitpunktzahl)	32	IEEE Gleitpunktzahl	Obere Grenze: ±3.402823e+38 Untere Grenze: ±1.175 495e-38	L 1.234567e+13

Typ und Beschreibung	Größe in Bits	Formatoptionen	Bereich und Zahlendarstellung (niedrigster bis höchster Wert)	Beispiel
S5TIME (SIMATIC-Zeit)	16	S7-Zeit in Schritten von 10 ms (Default-Wert)	S5T#0H_0M_0S_10MS bis S5T#2H_46M_30S_0MS und S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC-Zeit)	32	IEC-Zeit in Schritten von 1 ms, Ganzzahl mit Vorzeichen	- T#24D_20H_31M_23S_648MS bis T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (IEC-Datum)	16	IEC-Datum in Schritten von 1 Tag	D#1990-1-1 bis D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (Uhrzeit)	32	Uhrzeit in Schritten von 1 ms	TOD#0:0:0.0 bis TOD #23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Zeichen)	8	ASCII-Zeichen	'A','B' usw.	L 'E'

A.3.2.1 Format des Datentyps INT (16-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, ob es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (16 Bit) im Speicher belegt, beträgt ein Wort. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (16 Bit).

Format	Bereich
Ganzzahl (16 Bit)	–32 768 bis +32 767

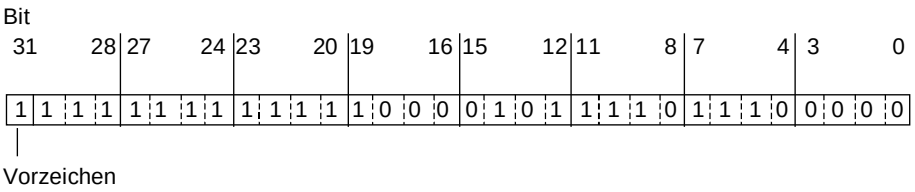
Nachfolgendes Bild zeigt die Ganzzahl +44 als Dualzahl.

A.3.2.2 Format des Datentyps DINT (32-Bit-Ganzzahlen)

Eine Ganzzahl hat ein Vorzeichen, das angibt, ob es sich um eine positive oder negative ganze Zahl handelt. Der Platz, den eine Ganzzahl (32 Bit) im Speicher belegt, beträgt zwei Wörter. Nachfolgende Tabelle zeigt den Bereich einer Ganzzahl (32 Bit).

Format	Bereich
Ganzzahl (32 Bit)	–2 147 483 648 bis +2 147 483 647

Nachfolgendes Bild zeigt die Ganzzahl –500 000 als Dualzahl. Im Dualsystem wird die negative Form einer Ganzzahl als Zweierkomplement der positiven Ganzzahl dargestellt. Das Zweierkomplement einer Ganzzahl erhalten Sie, indem Sie die Signalzustände aller Bits umkehren und dann +1 zum Ergebnis addieren.



A.3.2.3 Format des Datentyps REAL (Gleitpunktzahlen)

Zahlen im Gleitpunktformat werden in der allgemeinen Form "Zahl = m ∗ b hoch E" dargestellt. Die Basis ”b” und der Exponent ”E” sind ganze Zahlen, die Mantisse ”m” eine rationale Zahl.

Diese Zahlendarstellung hat den Vorteil, daß auf begrenztem Raum sehr große und sehr kleine Werte dargestellt werden können. Mit der begrenzten Anzahl von Bits für Mantisse und Exponent läßt sich ein weiter Zahlenbereich abdecken.

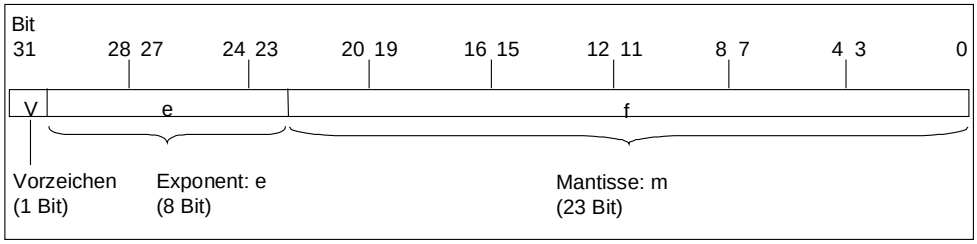
Der Nachteil liegt in der begrenzten Rechengenauigkeit: Beispielsweise müssen bei der Summenbildung zweier Zahlen die Exponenten durch Verschieben der Mantisse (gleitender Dezimalpunkt) angeglichen werden (Addition der Mantissen zweier Zahlen mit gleichem Exponent).

Gleitpunktformat in STEP 7

Gleitpunktzahlen in STEP 7 entsprechen dem Grundformat mit einfacher Breite, wie in der Norm ANSI/IEEE Standard 754–1985, *IEEE Standard for Binary Floating-Point Arithmetic* beschrieben. Sie bestehen aus den folgenden Komponenten:

- dem Vorzeichen V
- dem um eine Konstante (Bias = +127) erhöhten Exponenten e = E + Bias
- dem gebrochenen Teil der Mantisse m.
  Der ganzzahlige Anteil der Mantisse wird nicht mit abgelegt, da er innerhalb des gültigen Zahlenbereichs immer = 1 ist

Die drei Bestandteile belegen zusammen ein Doppelwort (32 Bit):



Die folgende Tabelle zeigt die Wertigkeit der einzelnen Bits im Gleitpunktformat.

Bestandteil der Gleitpunktzahl	**Bit-Nummer**	**Wertigkeit**
Vorzeichen V	31	
Exponent **e**	30	2 hoch 7
...	...	...
Exponent **e**	24	2 hoch 1
Exponent **e**	23	2 hoch 0
Mantisse **m**	22	2 hoch –1
...	...	...
Mantisse **m**	1	2 hoch –22
Mantisse **m**	0	2 hoch –23

Mit den drei Komponenten **V**, **e** und **m** ist der Wert einer in diesem Format dargestellten Zahl definiert über die Formel:

Zahl = 1.**m** ∗ 2 hoch (e-Bias)

Dabei gilt:

- e: $1 \leq e \leq 254$
- Bias: Bias = 127. Somit entfällt ein extra Vorzeichen für den Exponent.
- V: für eine positive Zahl ist V = 0 und für negative Zahl ist V = 1.

Wertebereich der Gleitpunkzahlen

Anhand des oben dargestellten Gleitpunktformats ergeben sich die

- kleinste Gleitpunktzahl = 1.0 ∗ 2 hoch (1-127) = 1.0 ∗ 2 hoch (-126) = 1.175 495E–38 und
- größte Gleitpunktzahl = 2-2 hoch (-23) ∗ 2 hoch (254-127) = 2-2 hoch (-23) ∗ 2 hoch (+127) = 3.402 823E+38

Die Zahl Null wird mit e = m = 0 dargestellt; e = 255 und m = 0 steht für ”unendlich”.

Format	**Bereich[1)]**
Gleitpunktzahlen nach ANSI/IEEE Standard	–3.402 823E+38 bis –1.175 495E–38 und 0 und +1.175 495E–38 bis +3.402 823E+38

Die nächste Tabelle zeigt den Signalzustand der Bits im Statuswort für die Ergebnisse von Operationen mit Gleitpunktzahlen, die nicht innerhalb des gültigen Bereichs liegen.

Ungültiger Bereich für ein Ergebnis	**A1**	**A0**	**OV**	**OS**
-1.175494E-38 < Ergebnis < -1.401298E-45 (negative Zahl) Unterschreitung	0	0	1	1
+1.401298E-45 < Ergebnis < +1.175494E-38 (positive Zahl) Unterschreitung	0	0	1	1
Ergebnis < -3.402823E+38 (negative Zahl) Überlauf	0	1	1	1
Ergebnis > 3.402823E+38 (positive Zahl) Überlauf	1	0	1	1
keine gültige Gleitpunktzahl oder unzulässige Operation (Eingangswert außerhalb des gültigen Wertebereichs)	1	1	1	1

Vorsicht bei mathematischen Operationen:

Das Ergebnis "Keine gültige Gleitpunktzahl" erhält man beispielsweise bei dem Versuch, die Quadratwurzel aus –2 zu ziehen. Werten Sie deshalb bei mathematischen Operationen stets zuerst die Statusbits aus, bevor Sie mit dem Ergebnis weiterrechnen.

Vorsicht beim "Steuern von Variablen":

Legt man die Werte für Gleitkommaoperationen z.B. in Merkerdoppelwörtern ab, so kann man diese Werte mit beliebigen Bitmustern verändern. Jedoch ist nicht jedes Bitmuster eine gültige Zahl!

Genauigkeit beim Rechnen mit Gleitpunktzahlen

Vorsicht

Bei umfangreichen Berechnungen mit Zahlen, die große Größenunterschiede (mehrere 10-er Potenzen) aufweisen, können Ungenauigkeiten im Rechenergebnis entstehen.

Die Gleitpunktzahlen in STEP 7 sind auf 6 Dezimalstellen genau. Sie können deshalb bei der Angabe von Gleitpunktkonstanten nur maximal 6 Nachkommastellen angeben.

Hinweis

Die Rechengenauigkeit von 6 Dezimalstellen bedeutet z.B., daß die Addition von Zahl1 + Zahl2 = Zahl1 ergibt, wenn Zahl1 größer Zahl2 ∗ 10 hoch y, mit y>6 ist:

100 000 000 + 1 = 100 000 000.

Beispiele für Zahlen in Gleitpunktdarstellung

Nachfolgendes Bild zeigt das Gleitpunktzahlenformat für die folgenden dezimalen Werte:

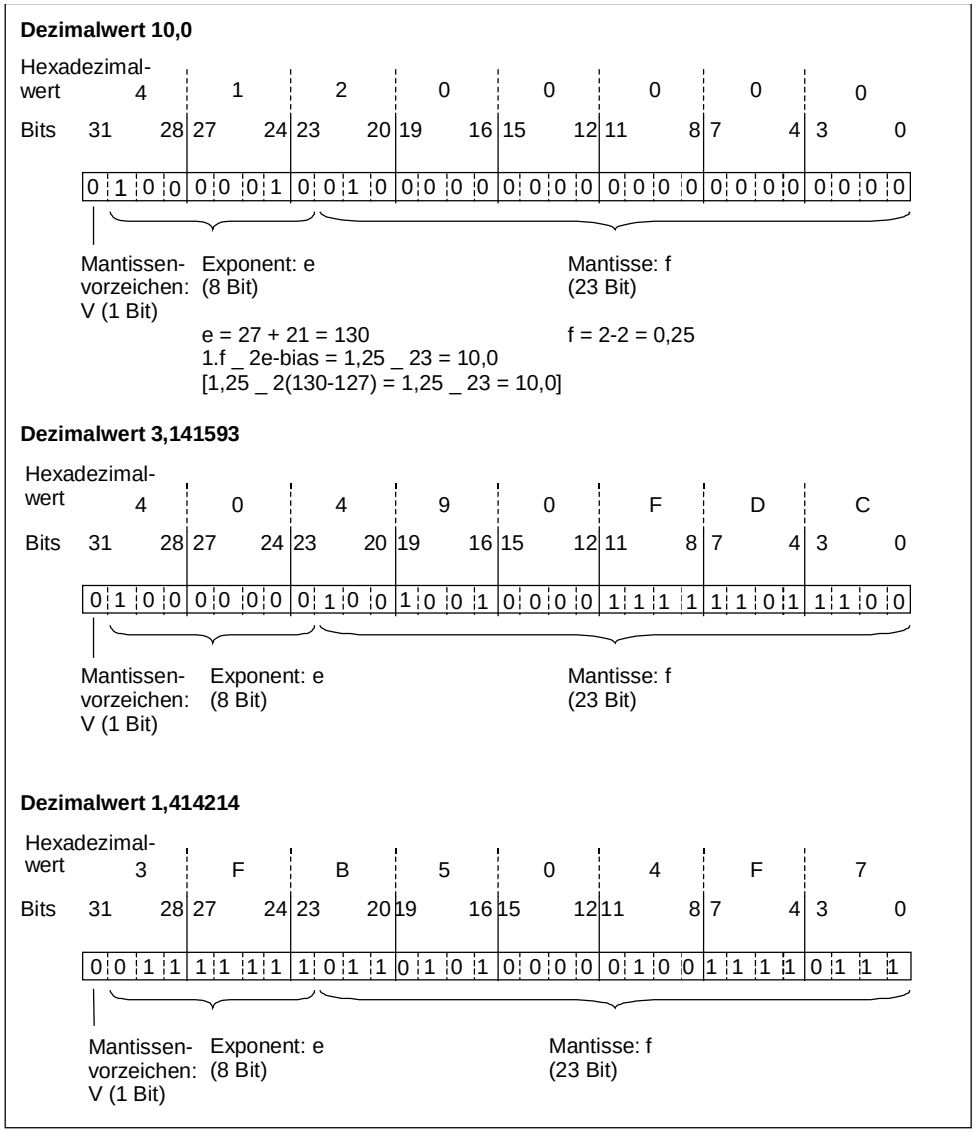
- 10,0
- p (3,141593)
- Quadratwurzel aus 2 (p2 = 1,414214)

Die Zahl **10.0** im ersten Beispiel ergibt sich aus ihrem Gleitpunktformat (Darstellung in HEX: 4120 0000) wie folgt:

e = 2 hoch 1 + 2 hoch 7 = 2 + 128 = 130

m = 2 hoch (-2) = 0,25

Damit ergibt sich: 1.m ∗ 2 hoch (e – Bias) = 1.25 ∗ 2 hoch (130 – 127) = 1.25 ∗ 2 hoch 3 = 10.0.



A.3.2.4 Format der Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen

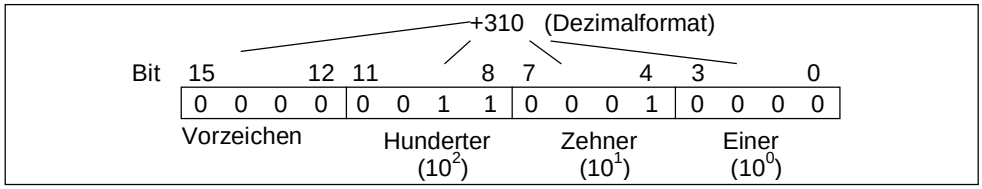
Die binär-codierte Darstellung (BCD) stellt eine Dezimalzahl durch Gruppen von Binärziffern (Bits) dar. Eine Gruppe von 4 Bits stellt eine Ziffer einer Dezimalzahl oder das Vorzeichen der Dezimalzahl dar. Die Gruppen von 4 Bits bilden ein Wort (16 Bits) oder Doppelwort (32 Bits). Die vier höchstwertigen Bits geben das Vorzeichen der Zahl an (”1111” bedeutet Minus und ”0000” Plus). Befehle mit BCD-codierten Operanden werten nur das höchstwertige Bit (15 bei Wort-, 31 bei Doppelwortformat) aus.

Nachfolgende Tabelle zeigt Format und Bereich für die beiden Arten der BCD-Zahlen.

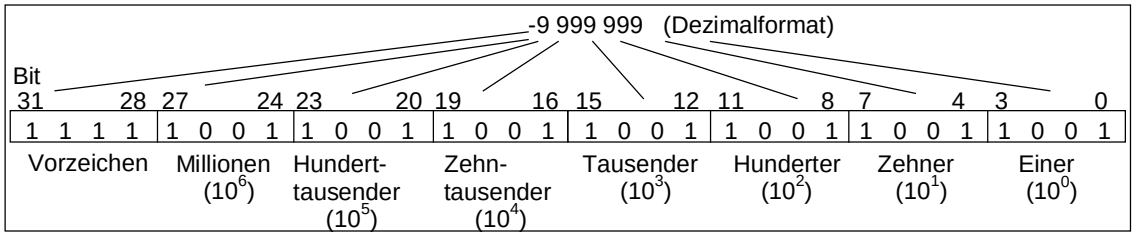
Format	Bereich
Wort (16 Bits, 3stellige BCD-Zahl mit Vorzeichen)	–999 bis +999
Doppelwort (32 Bits, 7stellige BCD-Zahl mit Vorzeichen)	–9 999 999 bis +9 999 999

Die nachfolgenden Bilder zeigen Beispiele für eine binär-codierte Dezimalzahl in folgenden Formaten:

- Wortformat

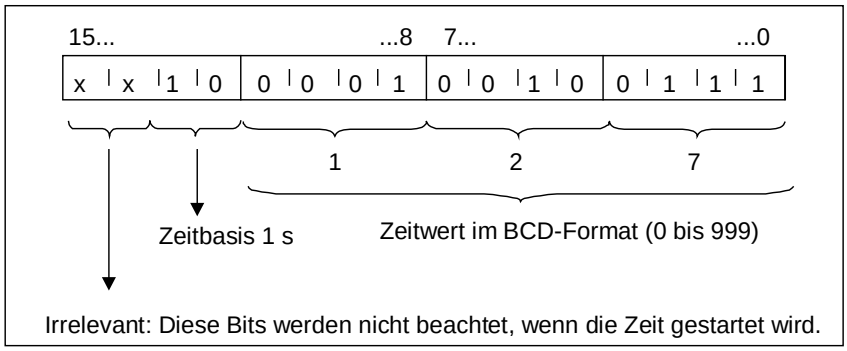


- Doppelwortformat


-9 999 999 (Dezimalformat)
Bit
31 28 27 24 23 20 19 16 15 12 11 8 7 4 3 0
1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1
Vorzeichen
Millionen (10^6)
Hundert-tausender (10^5)
Zehn-tausender (10^4)
Tausender (10^3)
Hunderter (10^2)
Zehner (10^1)
Einer (10^0)

A.3.2.5 Format des Datentyps S5TIME (Zeitdauer)

Wenn Sie die Zeitdauer mit dem Datentyp S5TIME eingeben, werden Ihre Eingaben im BCD-Format gespeichert. Nachfolgendes Bild zeigt den Inhalt des Zeitoperanden bei einem Zeitwert von 127 und einer Zeitbasis von 1 s.



Wenn Sie mit S5TIME arbeiten, geben Sie einen Zeitwert im Bereich von 0 bis 999 ein und bezeichnen eine Zeitbasis (siehe nachfolgende Tabelle). Die Zeitbasis gibt das Intervall an, in dem eine Zeit den Zeitwert um je eine Einheit vermindert, bis er ”0” erreicht.

Zeitbasis für S5TIME

Zeitbasis	Binärcode für Zeitbasis
10 ms	00
100 ms	01
1 s	10
10 s	11

Mit der folgenden Syntax können Sie einen vordefinierten Zeitwert laden:

- L[1)] W#16#wxyz
  - mit: w = Zeitbasis (d. h. Zeitintervall oder Auflösung)
  - xyz = Zeitwert im BCD-Format
- L[1)] S5T#aH_bbM_ccS_dddMS
  - mit: a = Stunden, bb = Minuten, cc = Sekunden und ddd = Millisekunden.
  - Die Zeitbasis wird automatisch gewählt und der Wert zur nächstniederen Zahl mit dieser Zeitbasis gerundet.

Sie können einen Zeitwert von max. 9 990 Sekunden bzw. 2H_46M_30S eingeben.

[1)] = L nur bei AWL-Programmierung angeben

A.3.3 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen definieren Datengruppen, die größer sind als 32 Bits bzw. Datengruppen, die sich aus anderen Datentypen zusammensetzen. STEP 7 läßt die folgenden zusammengesetzten Datentypen zu:

- DATE_AND_TIME
- STRING
- ARRAY (Feld)
- STRUCT (Struktur)
- UDT (Anwenderdefinierte Datentypen)
- FB und SFB

Nachfolgende Tabelle beschreibt die zusammengesetzten Datentypen. Sie definieren Strukturen und Felder entweder in der Variablendeklaration des Codebausteins oder in einem Datenbaustein.

Datentyp	Beschreibung
DATE_AND_TIME DT	Definiert einen Bereich mit 64 Bits (8 Bytes). Dieser Datentyp speichert in binärcodiertem Dezimalformat.
STRING	Definiert eine Gruppe von maximal 254 Zeichen (Datentyp CHAR). Der Standardbereich, der für einen Zeichenkette reserviert ist, besteht aus 256 Bytes. Dies ist der Platz, der benötigt wird, um 254 Zeichen und einen Kopf von 2 Bytes zu speichern. Sie können den Speicherplatz für eine Zeichenkette verringern, indem Sie auch die Anzahl der Zeichen definieren, die in der Zeichenkette gespeichert werden sollen (z. B.: string[9] 'Siemens').
ARRAY	Definiert eine multidimensionale Gruppierung eines Datentyps (entweder elementar oder zusammengesetzt). Zum Beispiel: "ARRAY [1..2,1..3] OF INT" definiert ein Feld im Format 2 x 3 aus Ganzzahlen. Sie greifen auf die Daten, die in einem Feld gespeichert sind, über den Index ("[2,2]") zu. Sie können maximal 6 Dimensionen in einem Feld definieren. Der Index kann ein beliebiger ganzzahliger Wert (-32768 bis 32767) sein.
STRUCT	Definiert eine Gruppierung von beliebig kombinierten Datentypen. Sie können z. B. ein Feld aus Strukturen oder eine Struktur aus Strukturen und Feldern definieren.
UDT	Ermöglicht es, große Datenmengen zu strukturieren und das Eingeben der Datentypen zu vereinfachen, wenn Datenbausteine erzeugt oder Variablen in der Variablendeklaration deklariert werden sollen. In STEP 7 können Sie zusammengesetzte und elementare Datentypen verknüpfen und so Ihren eigenen "anwenderdefinierten" Datentyp erstellen. UDTs haben einen eigenen Namen und sind deshalb mehrfach verwendbar.
FB, SFB	Bestimmen die Struktur des zugeordneten Instanz-Datenbausteins und ermöglichen die Übergabe von Instanzdaten für mehrere FB-Aufrufe in einem Instanz-DB.

Strukturierte Datentypen werden auf Wortgrenzen ausgerichtet (WORD aligned) abgelegt.

A.3.3.1 Format des Datentyps DATE_AND_TIME (Datum und Uhrzeit)

Wenn Sie Datum und Uhrzeit mit dem Datentyp DATE_AND_TIME (DT) eingeben, werden Ihre Eingaben im BCD-Format in 8 Bytes gespeichert. Der Datentyp DATE_AND_TIME umfaßt den folgenden Bereich:

DT#1990-1-1-0:0:0.0 bis DT#2089-12-31-23:59:59.999

Die folgenden Beispiele zeigen die Syntax, mit der Sie das Datum und die Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:01 Uhr und 1,23 Sekunden eingeben. Die folgenden beiden Formate sind möglich:

- DATE_AND_TIME#1993–12–25–8:01:1.23
- DT#1993-12-25-8:01:1.23

Die folgenden IEC-Standardfunktionen (International Electrotechnical Commission) stehen für das Arbeiten mit dem Datentyp DATE_AND_TIME zur Verfügung:

- Umwandeln des Datums und der Uhrzeit in das Format DATE_AND_TIME

  FC3: D_TOD_DT
- Herauslösen des Datums aus dem Format DATE_AND_TIME

  FC6: DT_DATE
- Herauslösen des Wochentags aus dem Format DATE_AND_TIME

  FC7: DT_DAY
- Herauslösen der Uhrzeit aus dem Format DATE_AND_TIME

  FC8: DT_TOD

In nachfolgender Tabelle ist der Inhalt der Bytes aufgeführt, die die Information zu Datum und Uhrzeit enthalten. Das Beispiel zeigt Datum und Uhrzeit für Donnerstag, den 25. Dezember 1993, 8:01 Uhr und 1,23 Sekunden.

Byte	Inhalt	Beispiel
0	Jahr	B#16#93
1	Monat	B#16#12
2	Tag	B#16#25
3	Stunde	B#16#08
4	Minute	B#16#01
5	Sekunde	B#16#01
6	Die beiden höchstwertigen Ziffern von MSEC	B#16#23
7 (4MSB)	Die niederwertigen Ziffern von MSEC	B#16#0
7 (4LSB)	Wochentag 1 = Sonntag 2 = Montag ... 7 = Samstag	B#16#5

Der erlaubte Bereich für den Datentyp ”DATE_AND_TIME” ist:

- min.: DT#1990-1-1-0:0:0.0
- max.: DT#2089-12-31-23:59:59.999

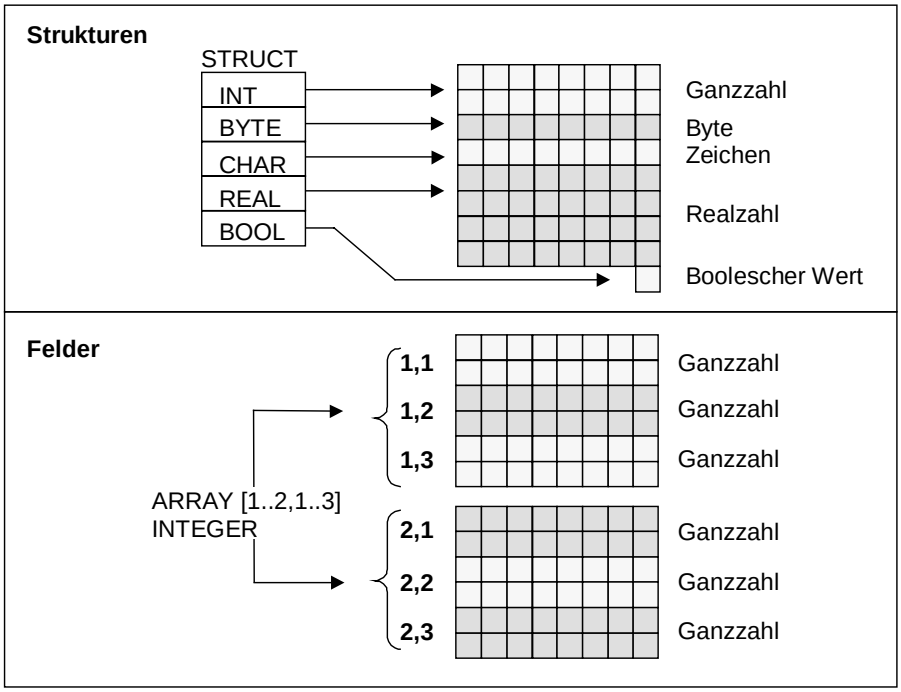
	Möglicher Wertebereich	**BCD-Code**
Jahr	1990 – 1999 2000 – 2089	90h – 99h 00h – 89h
Monat	1 – 12	01h – 12h
Tag	1 – 31	01h – 31h
Stunde	00 – 23	00h – 23h
Minute	00 – 59	00h – 59h
Sekunde	00 – 59	00h – 59h
Millisekunde	0 – 999	000h – 999h
Wochentag	Sonntag – Samstag	1h – 7h

A.3.3.2 Verwenden von zusammengesetzten Datentypen

Sie können neue Datentypen erstellen, indem Sie elementare und zusammengesetzte Datentypen zu den folgenden zusammengesetzten Datentypen verbinden:

- Feld (Datentyp ARRAY): Ein Feld verknüpft eine Gruppe eines Datentyps zu einer Einheit.
- Struktur (Datentyp STRUCT): Eine Struktur verknüpft verschiedene Datentypen zu einer Einheit.
- Zeichenkette (Datentyp STRING): Eine Zeichenkette definiert ein eindimensionales Feld mit maximal 254 Zeichen (Datentyp CHAR). Eine Zeichenkette kann nur als Einheit übergeben werden. Die Länge der Zeichenkette muß bei dem Formal- und dem Aktualparameter des Bausteins übereinstimmen.
- Datum und Zeit (Datentyp DATE_AND_TIME): Das Datum und die Zeit speichern Jahr, Monat, Tag, Stunden, Minuten, Sekunden, Millisekunden und Wochentag.

Nachfolgendes Bild zeigt, wie Felder und Strukturen Datentypen in einem Bereich strukturieren, um so Informationen zu speichern. Sie definieren ein Feld oder eine Struktur entweder in einem DB oder in der Variablendeklaration eines FB, OB oder einer FC.



A.3.3.3 Verwenden von Feldern für den Datenzugriff

Felder

Ein Feld verknüpft eine Gruppe eines Datentyps (elementar oder zusammengesetzt) zu einer Einheit. Sie können kein Feld aus Feldern erstellen. Wenn Sie ein Feld definieren, müssen Sie folgendes angeben:

- Sie geben dem Feld einen Namen.
- Sie deklarieren ein Feld mit dem Schlüsselwort ARRAY.
- Sie geben die Größe des Felds mit Hilfe eines Index an. Sie geben die erste und die letzte Zahl der einzelnen Dimensionen (maximal 6) in dem Feld an. Den Index geben Sie in eckigen Klammern an, wobei jede Dimension durch Komma getrennt und die erste und die letzte Zahl der Dimension durch zwei Punkte voneinander getrennt werden. Der folgende Index beispielsweise definiert ein dreidimensionales Feld:

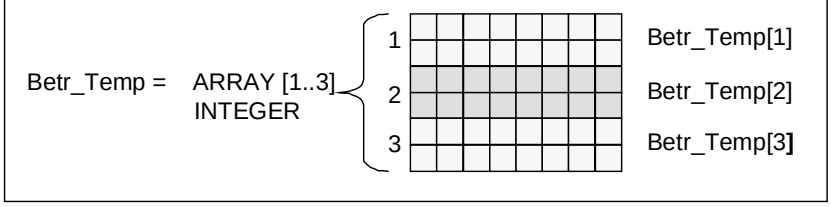
  [1..5,–2..3,30..32]
- Sie geben den Datentyp der Daten an, die in dem Feld gespeichert werden sollen.

Beispiel 1

Nachfolgendes Bild zeigt ein Feld aus drei Ganzzahlen. Sie greifen über den Index auf die Daten zu, die in einem Feld gespeichert sind. Bei dem Index handelt es sich um die Zahl in eckigen Klammern. Der Index der zweiten Ganzzahl beispielsweise ist Betr_Temp[2].

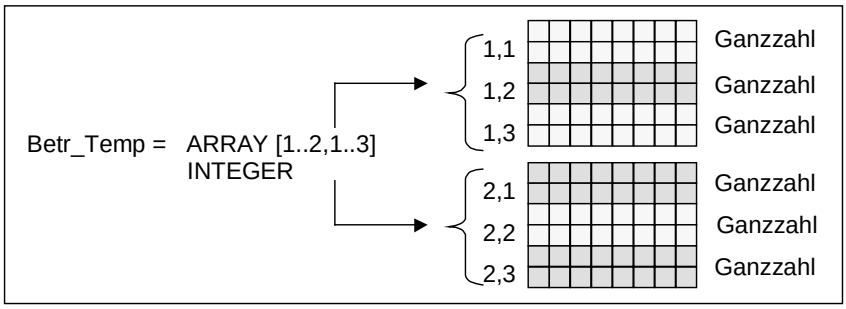
Ein Index kann ein beliebiger ganzzahliger Wert (-32768 bis 32767) sein, einschließlich negativer Werte. Das Feld im nachfolgenden Bild könnte auch als ARRAY [-1..1] definiert werden. Der Index der ersten Ganzzahl wäre dann Betr_Temp[-1], der zweiten Betr_Temp[0] und der dritten Ganzzahl Betr_Temp[1].

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Betr_Temp	ARRAY[1..3]		
*2.0		INT		
=3.0		END_STRUCT		



Beispiel 2

Ein Feld kann auch eine multidimensionale Gruppe von Datentypen beschreiben. Nachfolgendes Bild zeigt ein zweidimensionales Feld aus Ganzzahlen.



Sie greifen auf die Daten in einem multidimensionalen Feld über den Index zu. Bei dem Beispiel ist die erste Ganzzahl Betr_Temp[1,1], die dritte ist Betr_Temp[1,3], die vierte ist Betr_Temp[2,1] und die sechste ist Betr_Temp[2,3].

Sie können maximal sechs Dimensionen (sechs Indizes) für ein Feld definieren. Sie könnten beispielsweise die Variable Betr_Temp folgendermaßen als sechsdimensionales Feld definieren:

ARRAY [1..3,1..2,1..3,1..4,1..3,1..4]

Der Index des ersten Elements in diesem Feld ist Betr_Temp[1,1,1,1,1,1]. Der Index des letzten Elements ist Betr_Temp[3,2,3,4,3,4].

Erstellen von Feldern

Sie definieren Felder, wenn Sie Daten in einem DB oder in der Variablendeklaration deklarieren. Wenn Sie das Feld deklarieren, dann geben Sie das Schlüsselwort (ARRAY) an, danach die Größe in eckigen Klammern:

[unterer Grenzwert..oberer Grenzwert]

Bei einem multidimensionalen Feld geben Sie die zusätzlichen oberen und unteren Grenzwerte an und trennen die einzelnen Dimensionen durch ein Komma. Nachfolgendes Bild zeigt die Deklaration zum Erstellen eines Felds im Format 2 x 3.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Waerme_2x3	ARRAY[1..2,1..3]		
*2.0		INT		
=6.0		END_STRUCT		

Eingeben von Anfangswerten für ein Feld

Sie können beim Erstellen der Felder jedem Feldelement einen Anfangswert zuordnen. STEP 7 bietet Ihnen zwei Möglichkeiten zum Eingeben von Anfangswerten:

- Eingeben von individuellen Werten: Für jedes Element des Felds geben Sie einen gültigen Wert an (für den Datentyp des Felds). Geben Sie die Werte in der Reihenfolge der Elemente an: [1,1]. Beachten Sie, daß Sie die einzelnen Elemente durch Komma voneinander trennen müssen.
- Angeben eines Wiederholfaktors: Bei sequentiellen Elementen, die über den gleichen Anfangswert verfügen, können Sie die Anzahl der Elemente (den Wiederholfaktor) und den Anfangswert für diese Elemente angeben. Das Format zum Eingeben des Wiederholfaktors ist *x(y)*, wobei *x* der Wiederholfaktor und *y* der Wert ist, der wiederholt werden soll.

Wenn Sie das Feld verwenden, das im vorherigen Bild deklariert wird, können Sie den Anfangswert für alle sechs Elemente folgendermaßen angeben: 17, 23, -45, 556, 3342, 0. Sie könnten den Anfangswert von allen sechs Elementen auch auf 10 setzen, indem Sie 6(10) angeben. Sie könnten für die ersten beiden Elemente bestimmte Werte angeben und dann die übrigen vier Elemente auf 0 setzen, indem Sie 17, 23, 4(0) angeben.

Zugreifen auf Daten in einem Feld

Sie greifen auf die Daten in einem Feld über den Index des bestimmten Elements in dem Feld zu. Der Index wird mit dem symbolischen Namen verwendet.

Beispiel: Wenn das im vorherigen Bild deklarierte Feld an dem ersten Byte von DB20 (Motor) beginnt, greifen Sie auf das zweite Element in dem Feld über folgende Adresse zu:

Motor.Waerme_2x3[1,2].

Verwenden von Feldern als Parameter

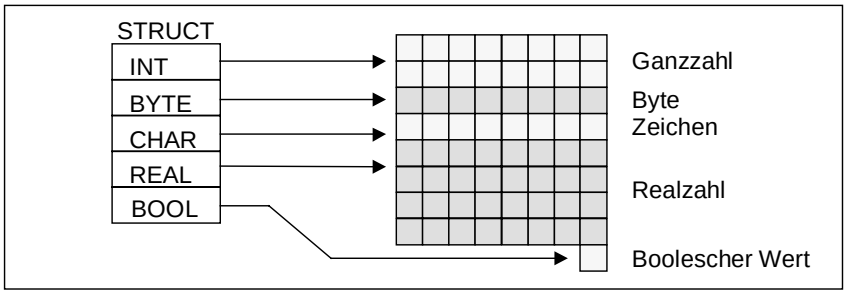
Sie können Felder als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als ARRAY deklariert ist, müssen Sie das ganze Feld übergeben (und nicht nur einzelne Elemente). Ein Element eines Felds kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element des Felds dem Datentyp des Parameters entspricht.

Wenn Sie Felder als Parameter verwenden, müssen die Felder nicht den gleichen Namen haben (sie brauchen gar keinen Namen zu haben). Beide Felder (der Formalparameter und der Aktualparameter) müssen aber die gleiche Struktur vorweisen. Ein Feld im Format 2 x 3 aus Ganzzahlen zum Beispiel kann als Parameter nur übergeben werden, wenn der Formalparameter des Bausteins ein Feld im Format 2 x 3 aus Ganzzahlen definiert und auch der Aktualparameter, der über die Aufrufoperation bereitgestellt wird, ein Feld im Format 2 x 3 aus Ganzzahlen ist.

A.3.3.4 Verwenden von Strukturen für den Datenzugriff

Strukturen

Eine Struktur verknüpft verschiedene Datentypen (elementare und zusammengesetzte Datentypen, einschließlich Felder und Strukturen) zu einer Einheit. So können Sie Daten entsprechend der Steuerung für Ihren Prozeß gruppieren. Sie können so auch Parameter als eine Dateneinheit übergeben, und nicht als einzelne Elemente. Nachfolgendes Bild zeigt eine Struktur, die aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Eine Struktur kann in maximal 8 Ebenen geschachtelt sein (z. B. eine Struktur aus Strukturen, die Felder enthält).

Erstellen einer Struktur

Sie definieren Strukturen beim Deklarieren von Daten innerhalb eines DB oder in der Variablendeklaration eines Codebausteins.

Nachfolgendes Bild zeigt die Deklaration einer Struktur (*Stapel_1*), die aus folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercodes), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel_1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE		
+4.0	Steuercode	CHAR		
+6.0	Temperatur	REAL	120	
+8.1	Ende	BOOL	FALSE	
=10.0		END_STRUCT		

Zuordnen von Anfangswerten für eine Struktur

Wenn Sie jedem Element einer Struktur einen Anfangswert zuordnen möchten, geben Sie jeweils einen gültigen Wert für den Datentyp und den Namen des Elements ein. Sie können beispielsweise (der im vorherigen Bild deklarierten Struktur) die folgenden Anfangswerte zuordnen:

Menge	=	100
Ursprungsdaten	=	B#(0)
Steuercode	=	‘Z’
Temperatur	=	120
Ende	=	False

Speichern von und Zugreifen auf Daten in Strukturen

Sie greifen auf die einzelnen Elemente einer Struktur zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die *Menge DB20.DBW0* und die Adresse für die *Temperatur* ist *DB20.DBD6*).

Verwenden von Strukturen als Parameter

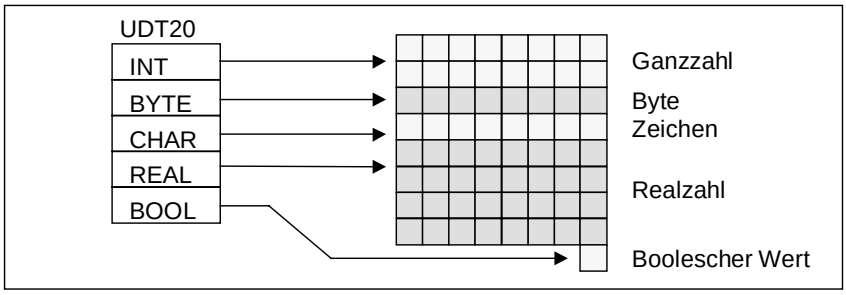
Sie können Strukturen als Parameter übergeben. Wenn ein Parameter in der Variablendeklaration als STRUCT deklariert ist, müssen Sie eine Struktur mit dem gleichen Aufbau übergeben. Ein Element einer Struktur kann jedoch einem Parameter zugeordnet werden, wenn Sie einen Baustein aufrufen, sofern das Element der Struktur dem Datentyp des Parameters entspricht.

Wenn Sie Strukturen als Parameter verwenden, müssen beide Strukturen (für den Formalparameter und den Aktualparameter) den gleichen Aufbau der Daten vorweisen, d. h. die gleichen Datentypen müssen in der gleichen Reihenfolge angeordnet sein.

A.3.3.5 Verwenden von anwenderdefinierten Datentypen für den Datenzugriff

Anwenderdefinierte Datentypen

Anwenderdefinierte Datentypen (UDT) können elementare und zusammengesetzte Datentypen verknüpfen. Sie können UDTs mit einem Namen versehen und mehrfach verwenden. Nachfolgendes Bild zeigt die Struktur eines UDT, der aus einer Ganzzahl, einem Byte, einem Zeichen, einer Gleitpunktzahl und einem booleschen Wert besteht.



Anstatt alle Datentypen einzeln oder als Struktur einzugeben, müssen Sie nur “UDT20” als Datentyp angeben und STEP 7 ordnet automatisch den entsprechenden Speicherplatz zu.

Erstellen eines anwenderdefinierten Datentyps

UDTs definieren Sie mit STEP 7. Nachfolgendes Bild zeigt einen UDT, der aus den folgenden Elementen besteht: eine Ganzzahl (zum Speichern der Menge), ein Byte (zum Speichern der Ursprungsdaten), ein Zeichen (zum Speichern des Steuercodes), eine Gleitpunktzahl (zum Speichern der Temperatur) und ein boolescher Merker (zum Beenden des Signals). Sie können dem UDT in der Symboltabelle einen symbolischen Namen zuordnen (z. B. *Prozeßdaten*).

Adresse	Name	Typ	Anfangswert	Kommentar
0.0	Stapel_1	STRUCT		
+0.0	Menge	INT	100	
+2.0	Ursprungsdaten	BYTE		
+4.0	Steuercode	CHAR		
+6.0	Temperatur	REAL	120	
+8.1	Ende	BOOL	FALSE	
=10.0		END_STRUCT		

Nachdem Sie einen UDT erstellt haben, können Sie den UDT wie einen Datentyp verwenden, so z. B. wenn Sie für eine Variable den Datentyp *UDT200* in einem DB deklarieren (oder in der Variablendeklaration eines FB).

Nachfolgendes Bild zeigt einen DB mit der Variablen *Prozessdaten_1* mit dem Datentyp UDT200. Sie geben nur *UDT200* und *Prozessdaten_1* an. Die kursiv dargestellten Felder werden erstellt, wenn Sie den DB übersetzen.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+6.0	Prozessdaten_**1**	UDT200		
=6.0		END_STRUCT		

Zuordnen von Anfangswerten für einen anwenderdefinierten Datentyp

Wenn Sie jedem Element eines UDT Anfangswerte zuordnen möchten, geben Sie für den Datentyp und den Namen eines jeden Elements einen gültigen Wert an. Sie können beispielsweise (dem im vorherigen Bild deklarierten UDT) die folgenden Anfangswerte zuordnen:

Menge	=	100
Ursprungsdaten	=	B#(0)
Steuercode	=	‘Z’
Temperatur	=	120
Ende	=	False

Wenn Sie eine Variable als UDT deklarieren, dann sind die Anfangswerte der Variablen die Werte, die Sie beim Erstellen des UDT angegeben haben.

Speichern von und Zugreifen auf Daten in einem anwenderdefinierten Datentyp

Sie greifen auf die einzelnen Elemente eines UDT zu. Sie können symbolische Adressen (z. B. *Stapel_1.Temperatur*) verwenden. Sie können aber auch die absolute Adresse angeben, unter der das Element gespeichert ist (Beispiel: ist *Stapel_1* in DB20 mit Beginn bei Byte 0 gespeichert, dann ist die absolute Adresse für die *Menge DB20.DBW0* und die Adresse für die *Temperatur* ist *DB20.DBD6*).

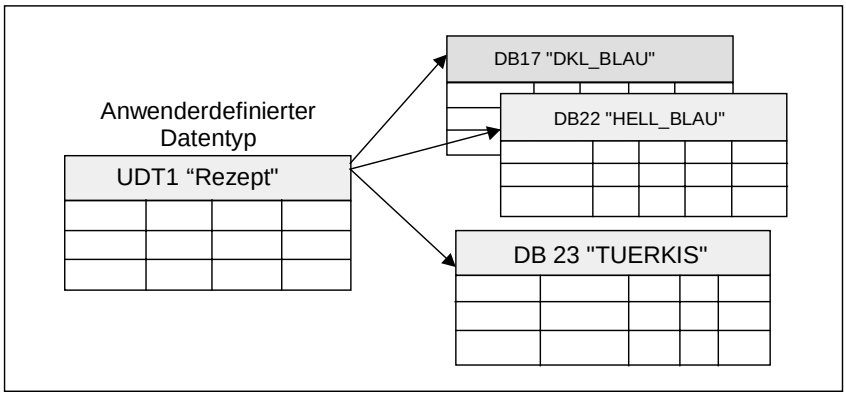
Verwenden von anwenderdefinierten Datentypen als Parameter

Sie können Variablen vom Datentyp UDT als Parameter übergeben. Wenn der Parameter in der Variablendeklaration als UDT deklariert ist, müssen Sie einen UDT übergeben, der die gleiche Struktur der Datenelemente hat. Ein Element eines UDT kann aber auch beim Aufrufen eines Bausteins einem Parameter zugeordnet werden, sofern das Element des UDT mit dem Datentyp des Parameters übereinstimmt.

Vorteile von DBs mit zugeordnetem UDT

Mit Hilfe von einmal erstellten UDTs können Sie eine Vielzahl von Datenbausteinen erzeugen, die die gleiche Datenstruktur haben. Diese Datenbausteine können von Ihnen durch Eingabe von unterschiedlichen Aktualwerten genau für die entsprechende Aufgabe angepaßt werden.

Strukturieren Sie z.B. einen UDT für ein Rezept (z.B. für das Mischen von Farben), dann können Sie diesem UDT mehrere DBs zuordnen, die jeweils die anderen Mengenangaben beinhalten:



Die Struktur des Datenbausteins wird durch den zugeordneten UDT festgelegt.

A.3.4 Parametertypen

Zusätzlich zu elementaren und zusammengesetzten Datentypen können Sie Parametertypen für Formalparameter definieren, die zwischen Bausteinen übergeben werden. STEP 7 kennt folgende Parametertypen:

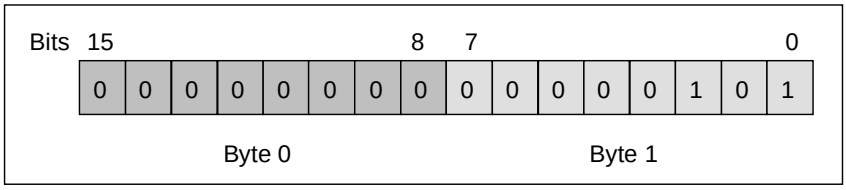
- TIMER oder COUNTER: legen eine bestimmte Zeit oder einen bestimmten Zähler fest, der bei der Bearbeitung verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp TIMER oder COUNTER versorgen, muß der zugehörige Aktualparameter eine Zeit oder ein Zähler sein, d. h. Sie geben ein “T” oder “Z” an, auf das eine positive Ganzzahl folgt.
- BLOCK: legt einen bestimmten Baustein fest, der als Eingang oder als Ausgang verwendet werden soll. Die Deklaration des Parameters bestimmt die Bausteinart (FB, FC, DB usw.), die verwendet werden soll. Wenn Sie einen Formalparameter vom Parametertyp BLOCK versorgen, geben Sie eine Bausteinadresse als Aktualparameter an. Beispiel: “FC101” (bei absoluter Adressierung) oder “Ventil” (bei symbolischer Adressierung).
- POINTER: referenziert die Adresse einer Variablen. Ein Pointer enthält eine Adresse anstelle eines Werts. Wenn Sie einen Formalparameter vom Parametertyp POINTER versorgen, geben Sie als Aktualparameter die Adresse an. In STEP 7 können Sie einen Pointer im Pointerformat oder einfach als Adresse (z. B. M 50.0) angeben. Beispiel für ein Pointerformat zum Adressieren der Daten, die in M 50.0 beginnen: `P#M50.0`
- ANY: wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist oder wenn ein beliebiger Datentyp eingesetzt werden kann. Nähere Informationen zum Parameter ANY entnehmen Sie den Abschnitten "Format des Parametertyps ANY bzw. Verwenden des Parametertyps ANY".

Ein Parametertyp kann auch ein anwenderdefinierter Datentyp (UDT) sein. Nähere Informationen zu UDTs entnehmen Sie dem Abschnitt "Verwenden von anwenderdefinierten Datentypen für den Datenzugriff".

Parameter	Größe	Beschreibung
TIMER	2 Bytes	Kennzeichnet eine bestimmte Zeit, die vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: `T1`
COUNTER	2 Bytes	Kennzeichnet einen bestimmten Zähler, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: `Z10`
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 Bytes	Kennzeichnet einen bestimmten Baustein, der vom Programm im aufgerufenen Codebaustein verwendet werden soll. Format: `FC101` `DB42`
POINTER	6 Bytes	Kennzeichnet die Adresse. Format: `P#M50.0`
ANY	10 Bytes	Wird verwendet, wenn der Datentyp des Aktualparameters nicht bekannt ist. Format: `P#M50.0 BYTE 10` ANY-Format bei Datentypen `P#M100.0 WORD 5` `L#1COUNTER 10` ANY-Format bei Parametertypen

A.3.4.1 Format der Parametertypen BLOCK, COUNTER, TIMER

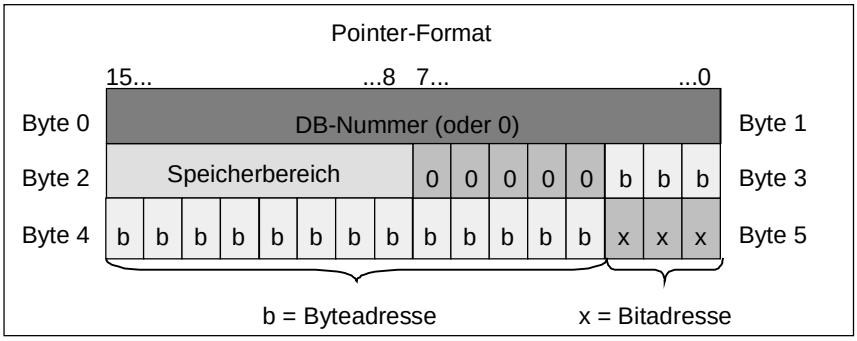
STEP 7 speichert die Parametertypen BLOCK, COUNTER und TIMER als Binärzahlen in einem Wort (32 Bits). Nachfolgendes Bild zeigt das Format dieser Parametertypen.



Die zulässige Anzahl der Bausteine, Zeiten und Zähler ist von Ihrer Variante der S7-CPU abhängig. Nähere Informationen zu der zulässigen Anzahl an Zeiten und Zählern sowie zu der maximalen Anzahl an verfügbaren Bausteinen entnehmen Sie den Datenblättern zu Ihrer CPU im Handbuch "Automatisierungssystem S7-300, Aufbau, CPU-Daten" oder im Installationshandbuch "Automatisierungssysteme S7-400/M7-400, Aufbauen".

A.3.4.2 Format des Parametertyps POINTER

STEP 7 speichert den Parametertyp POINTER in 6 Bytes (48 Bits). Nachfolgendes Bild zeigt die Art der Daten, die in jedem Byte gespeichert werden.



Der Parametertyp POINTER speichert die folgenden Informationen:

- DB-Nummer (bzw. 0, wenn die Daten nicht in einem DB gespeichert werden)
- Speicherbereich in der CPU (Nachfolgende Tabelle zeigt die Hexadezimalcodes der Speicherbereiche für den Parametertyp POINTER)

Hexadezimalcode	Speicherbereich	Beschreibung
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

- Adresse der Daten (im Format Byte.Bit)

  STEP 7 bietet das Pointer-Format: p#Speicherbereich Byte.Bit_Adresse. (Wenn der Formalparameter als Parametertyp POINTER deklariert wurde, dann brauchen Sie nur den Speicherbereich und die Adresse anzugeben. STEP 7 formatiert Ihren Eintrag dann automatisch in das Pointer-Format um.) Die folgenden Beispiele zeigen, wie Sie den Parametertyp POINTER für die Daten, die bei M50.0 beginnen, eingeben:

- P#M50.0
- M50.0 (wenn der Formalparameter als POINTER deklariert wurde)

A.3.4.3 Verwenden des Parametertyps POINTER

Ein Pointer wird verwendet, um auf einen Operanden zu zeigen. Der Vorteil dieser Adressierung ist, daß Sie den Operanden der Anweisung während der Programmbearbeitung dynamisch modifizieren können.

Pointer zur speicherindirekten Adressierung

Programmanweisungen, die mit der speicherindirekten Adressierung arbeiten, setzen sich zusammen aus einer Operation, einem Operandenkennzeichen und einem Versatz (der Versatz muß in eckigen Klammern angegeben werden).

Beispiel für einen Pointer im Doppelwortformat:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
T	MD2	Transferiere den Pointer nach MD2.
U	E [MD2]	Frage den Signalzustand an Eingang E 8.7 ab,
=	A [MD2]	und weise den Signalzustand Ausgang A 8.7 zu.

Pointer zur bereichsinternen und bereichsübergreifenden Adressierung

Die Programmanweisungen, die mit diesen Adressierungen arbeiten, setzen sich zusammen aus einer Operation und den folgenden Teilen:
Operandenkennzeichen, Adreßregister-Kennung, Versatz.

Das Adreßregister (AR1/2) und der Versatz müssen zusammen in eckigen Klammern angegeben werden.

Beispiel zur bereichsinternen Adressierung

Der Pointer enthält keine Angabe zu einem Speicherbereich:

L	P#8.7	Lade den Wert des Pointers in AKKU 1.
LAR1		Lade den Pointer von AKKU 1 in AR1.
U	E [AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	A [AR1, P#1.1]	weise den Signalzustand dem Ausgang zu A 10.0 zu

Der Versatz 0.0 hat keinen Einfluß. Ausgang 10.0 errechnet sich aus 8.7 (AR1) plus den Versatz 1.1. Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

Beispiel zur bereichsübergreifenden Adressierung

Bei der bereichsübergreifenden Adressierung ist im Pointer der Speicherbereich angegeben (im Beispiel E bzw. A).

L	P# E8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR1		Lade Speicherbereich E und die Adresse 8.7 in AR1.
L	P# A8.7	Lade den Wert des Pointers und Bereichskennung in AKKU 1.
LAR2		Lade Speicherbereich A und die Adresse 8.7 in AR2.
U	[AR1, P#0.0]	Frage den Signalzustand an Eingang E 8.7 ab und
=	[AR2, P#1.1]	weise den Signalzustand Ausgang A 10.0 zu.

Der Versatz 0.0 hat keinen Einfluß. Ausgang 10.0 errechnet sich aus 8.7 (AR2) plus 1.1 (Versatz). Das Ergebnis ist 10.0 und nicht 9.8, siehe Pointer-Format.

A.3.4.4 Baustein zum Verändern des Pointers

Mit Hilfe des Beispiel-Bausteins FC3 “Rangieren von Pointern” ist es möglich die Bit- oder Byteadresse eines Pointers zu verändern. Der zu verändernde Pointer wird beim FC-Aufruf an die Variable ”Zeiger” übergeben (es können bereichsinterne und bereichsübergreifende Pointer im Doppelwortformat verwendet werden).

Mit dem Parameter “Bit-Byte” können Sie die Bit- oder die Byteadresse des Pointers ändern (0: Bitadresse, 1: Byteadresse). Bei der Variable ”Inc_Wert” (im Format Integer) wird die Zahl angegeben, die zum Adreßinhalt addiert bzw. subtrahiert werden soll. Hier können Sie auch negative Zahlen zum Dekrementieren der Adresse angeben.

Bei Bitadressenänderung erfolgt ein Übertrag in die Byteadresse (auch bei Dekrementierung); z. B.:

- P#M 5.3, Bit_Byte = 0, Inc_Wert = 6 => P#M 6.1 oder
- P#M 5.3, Bit_Byte = 0, Inc_Wert = -6 => P#M 4.5.

Die Bereichsinformation des Pointers wird durch die Funktion nicht beeinflußt.

Der FC fängt einen Über-/Unterlauf des Zeigers ab. In diesem Fall wird der Zeiger nicht verändert und die ausgangsseitige Variable ”RET_VAL” (Fehlerbehandlung möglich) wird auf ”1” gesetzt (bis zur nächsten korrekten Bearbeitung der FC 3). Dies ist der Fall, wenn:

- 1.Bitadresse gewählt ist und Inc_Wert >7, oder <-7
- 2.Bit- oder Byteadresse gewählt ist und die Änderung eine ”negative” Byteadresse zur Folge hätte
- 3.Bit- oder Byteadresse gewählt ist und die Änderung eine unzulässig große Byteadresse zur Folge hätte.

Beispiel-Baustein in AWL zum Verändern des Pointers:

```
FUNCTION FC 3: BOOL
TITLE =Rangieren von Pointern
//Die FC 3 kann zum Verändern von Pointern verwendet werden.
AUTHOR : AUT1CS1
FAMILY : INDADR
NAME : ADRPOINT
VERSION : 0.0

VAR_INPUT
   Bit_Byte : BOOL ; //0: Bitadresse, 1: Byteadresse
   Inc_Wert : INT ; //Inkrement (wenn Wert neg.=> Dekrement/wenn Wert pos.
                    => Inkrement)
END_VAR

VAR_IN_OUT
   Zeiger : DWORD ; //der zu veränderten Pointer
END_VAR
VAR_TEMP
   Inc_Wert1 : INT ; //Zwischenwert Inkrement
   Zeiger1 : DWORD ; //Zwischenwert Zeiger
   Zw_Wert : DWORD ; //Hilfsvariable
END_VAR
BEGIN
NETWORK
TITLE =
//Der Baustein fängt Veränderungen, welche die Bereichsinformationen des
Zeigers
//verändern, oder zu ”negativen” Pointern führen automatisch ab!
   SET     ; //Setze VKE auf 1 und
   R       #RET_VAL; //setze Überlauf zurück
   L       #Zeiger; //Versorgen des temporären
   T       #Zeiger1; //Zwischenwertes Zeiger
   L       #Inc_Wert; //Versorgen des temporären
   T       #Inc_Wert1; //Zwischenwertes Inkrement
   U       #Bit_Byte; //wenn =1, dann Byteadressenoperation
```

```
      SPB   Byte; //Sprung zur Byteadressenberechnung
      L     7; //Wenn Wert des Inkrementes > 7,
      L     #Inc_Wert1;
      <I    ;
      S     #RET_VAL; //dann setze RET_VAL und
      SPB   Ende; //springe zum Ende
      L     -7; //Wenn Wert des Inkrementes < -7,
      <I    ;
      S     #RET_VAL; //dann setze RET_VAL und
      SPB   Ende; //springe zum Ende
      U     L     1.3; //wenn Bit 4 des Wertes = 1 (Inc_Wert neg) ist
      SPB   neg; //dann springe zu Bitadressen-Subtraktion
      L     #Zeiger1; //Lade Adressinformation des Zeigers
      L     #Inc_Wert1; //und addiere das Inkrement
      +D    ;
      SPA   test; //Springe zum Test auf negatives Ergebnis
neg:  L     #Zeiger1; //Lade Adressinformation des Zeigers
      L     #Inc_Wert1; //Lade das Inkrement
      NEGI  ; //negiere den negativen Wert,
      -D    ; //subtrahiere den Wert
      SPA   test; //und springe zum Test
Byte: L     0; //Beginn der Byteadressenänderung
      L     #Inc_Wert1; //Wenn Inkrement >=0, dann
      <I    ;
      SPB   pos; //springe zu Addition, sonst
      L     #Zeiger1; //lade Adressinformation des Zeigers,
      L     #Inc_Wert1; //lade das Inkrement,
      NEGI  ; //negiere den negativen Wert,
      SLD   3; //schiebe das Inkrement 3 Stellen nach links,
      –D    ; //subtrahiere den Wert
      SPA   test; //und springe zum Test
pos:  SLD   3; //schiebe Inkrement 3 Stellen nach links
      L     #Zeiger1; //lade Adressinformation des Zeigers
      +D    ; //addiere Inkrement
test: T     #Zw_Wert; //Transfer. Ergebnis Berechnungen in Zw_Wert,
      U     L     7.3; //Wenn ungültige Byteadresse (zu groß oder
```

```
      S     #RET_VAL; //negativ), dann setze RET_VAL
      SPB   Ende; //und springe zum Ende,
      L     #Zw_Wert; //sonst transferiere Ergebnis
      T     #Zeiger; //in Zeiger
Ende: NOP   0;
END_FUNCTION
```

A.3.4.5 Format des Parametertyps ANY

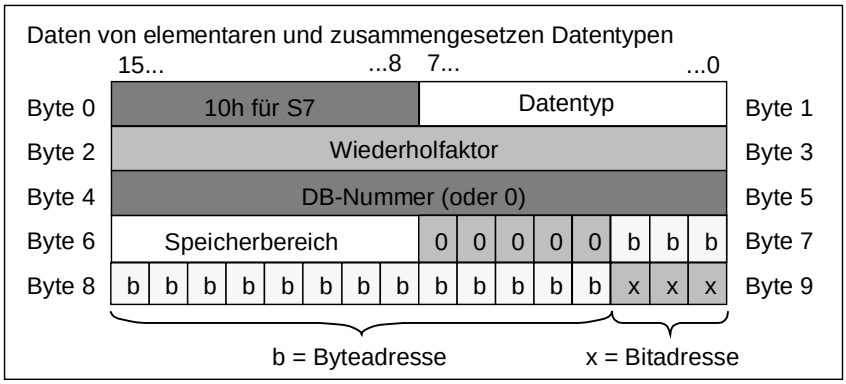
STEP 7 speichert die Daten des Parametertyps ANY in 10 Byte. Beim Aufbau eines Parameters vom Typ ANY müssen Sie darauf achten, daß alle 10 Byte belegt sind, da der aufgerufene Baustein den gesamten Inhalt des Parameters auswertet. Wenn Sie beispielsweise in Byte 4 eine DB-Nummer spezifizieren, müssen Sie auch den Speicherbereich in Byte 6 explizit angeben.

STEP 7 verwaltet die Daten von elementaren und zusammengesetzten Datentypen anders als die Daten für Parametertypen.

ANY-Format bei Datentypen

Bei elementaren und zusammengesetzten Datentypen speichert STEP 7 die folgenden Daten:

- Datentypen
- Wiederholfaktor
- DB-Nummer
- Speicherbereich, in dem die Informationen gespeichert werden
- Anfangsadresse der Daten



Der Wiederholfaktor kennzeichnet eine Menge des gekennzeichneten Datentyps zum Übergeben durch den Parametertyp ANY. Dadurch können Sie einen Datenbereich angeben und auch Felder und Strukturen in Verbindung mit dem Parametertyp ANY verwenden. STEP 7 kennzeichnet Felder und Strukturen als Anzahl (mit Hilfe des Wiederholfaktors) an Datentypen. Sollen z. B. 10 Wörter übergeben werden, muß beim Wiederholfaktor der Wert 10 und beim Datentyp der Wert 04 eingetragen sein.

Die Adresse wird im Format Byte.Bit gespeichert, wobei die Byteadresse in den Bits 0 bis 2 von Byte 7, in den Bits 0 bis 7 von Byte 8 und in den Bits 3 bis 7 von Byte 9 gespeichert wird. Die Bitadresse wird in den Bits 0 bis 2 von Byte 9 gespeichert.

Beim Nullpointer vom Datentyp NIL sind ab Byte 1 alle Bytes mit 0 belegt.

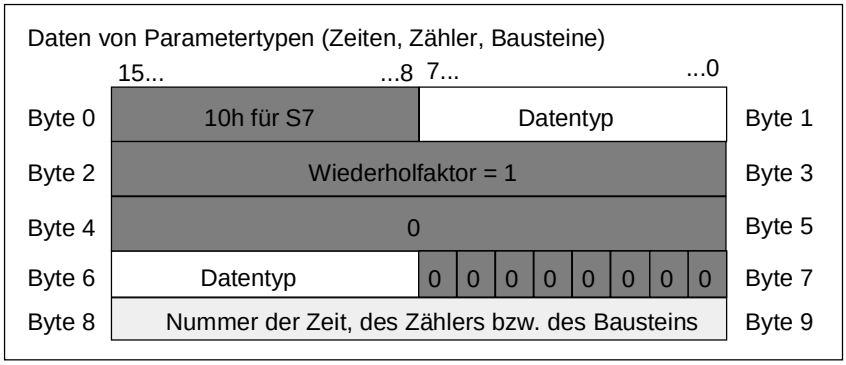
Nachfolgenden Tabellen zeigen die Codierung der Datentypen bzw. der Speicherbereiche für den Parametertyp ANY.

Codierung der Datentypen		
Hexadezimalcode	**Datentyp**	**Beschreibung**
b#16#00	NIL	Nullpointer
b#16#01	BOOL	Bits
b#16#02	BYTE	Bytes (8 Bits)
b#16#03	CHAR	Zeichen (8 Bits)
b#16#04	WORD	Wörter (16 Bits)
b#16#05	INT	Ganzzahlen (16 Bits)
b#16#06	DWORD	Wörter (32 Bits)
b#16#07	DINT	Ganzzahlen (32 Bits)
b#16#08	REAL	Gleitpunktzahlen (32 Bits)
b#16#09	DATE	Datum
b#16#0A	TIME_OF_DAY (TOD)	Uhrzeit
b#16#0B	TIME	Zeit
b#16#0C	S5TIME	Datentyp S5TIME
b#16#0E	DATE_AND_TIME (DT)	Datum und Zeit (64 Bits)
b#16#13	STRING	Zeichenkette

Codierung der Speicherbereiche		
Hexadezimalcode	**Bereich**	**Beschreibung**
b#16#81	E	Speicherbereich der Eingänge
b#16#82	A	Speicherbereich der Ausgänge
b#16#83	M	Speicherbereich der Merker
b#16#84	DB	Datenbaustein
b#16#85	DI	Instanz-Datenbaustein
b#16#86	L	Lokaldaten (L-Stack)
b#16#87	V	Vorherige Lokaldaten

ANY-Format bei Parametertypen

Bei Parametertypen speichert STEP 7 den Datentyp und die Adresse der Parameter. Der Wiederholfaktor ist immer 1. Byte 4, 5 und 7 sind immer 0. Byte 8 und 9 geben die Nummer der Zeit, des Zählers oder des Bausteins an.



Nachfolgende Tabelle zeigt die Codierung der Datentypen für den Parametertyp ANY bei Parametertypen.

Hexadezimalcode	Datentyp	Beschreibung
b#16#17	BLOCK_FB	FB-Nummer
b#16#18	BLOCK_FC	FC-Nummer
b#16#19	BLOCK_DB	DB-Nummer
b#16#1A	BLOCK_SDB	SDB-Nummer
b#16#1C	COUNTER	Nummer des Zählers
b#16#1D	TIMER	Nummer der Zeit

A.3.4.6 Verwenden des Parametertyps ANY

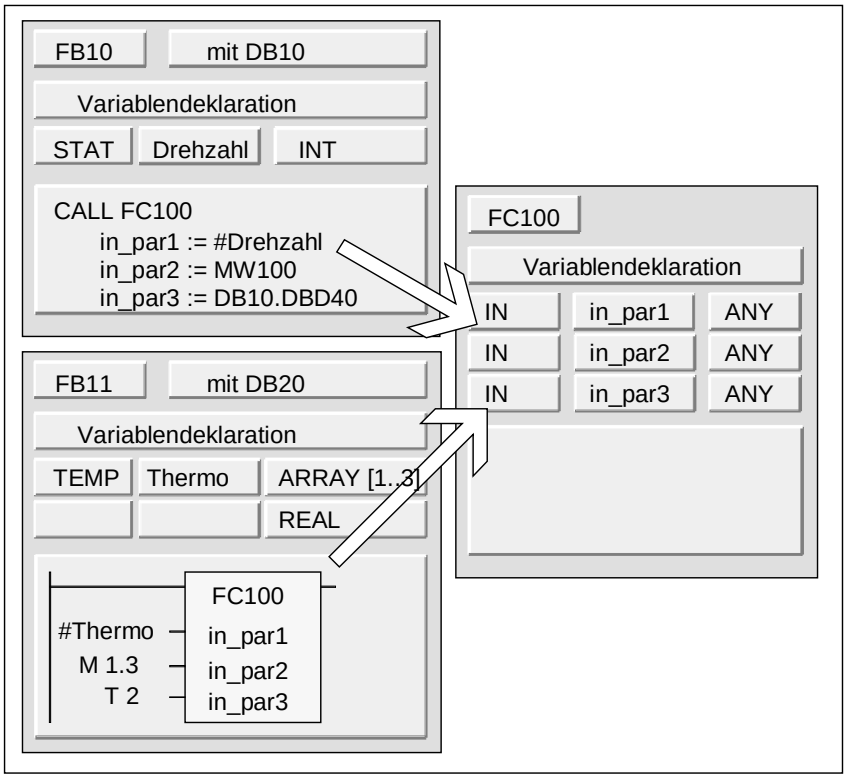
Sie können für einen Baustein Formalparameter definieren, die für Aktualparameter mit beliebigen Datentypen geeignet sind. Dies ist vor allem dann nützlich, wenn der Datentyp des Aktualparameters, der beim Aufrufen des Bausteins bereitgestellt wird, unbekannt ist oder variieren kann (und wenn ein beliebiger Datentyp zulässig ist). In der Variablendeklaration des Bausteins deklarieren Sie den Parameter als Datentyp ANY. In STEP 7 können Sie dann einen Aktualparameter eines beliebigen Datentyps zuordnen.

STEP 7 ordnet einer Variablen vom Datentyp ANY 80 Bits Speicherplatz zu. Wenn Sie diesem Formalparameter einen Aktualparameter zuordnen, codiert STEP 7 die Startadresse, den Datentyp und die Länge des Aktualparameters in den 80 Bits. Der aufgerufene Baustein analysiert die 80 Bits an Daten, die für den Parameter ANY gespeichert wurden, und erhält so Informationen, die für die zusätzliche Bearbeitung verwendet werden können.

Zuordnen eines Aktualparameters zu einem Parameter ANY

Wenn Sie für einen Parameter den Datentyp ANY deklarieren, können Sie diesem Formalparameter einen Aktualparameter mit beliebigem Datentyp zuordnen. In STEP 7 können Sie die folgenden Datentypen als Aktualparameter zuordnen:

- Elementare Datentypen: Sie geben die absolute Adresse oder den symbolischen Namen des Aktualparameters an.
- Zusammengesetzte Datentypen: Sie geben den symbolischen Namen der Daten mit zusammengesetztem Datentyp an (z. B. Felder und Strukturen).
- Zeiten, Zähler und Bausteine: Sie geben die Nummer an (z. B. T1, Z20 oder FB6).
- Nachfolgendes Bild zeigt, wie Daten an eine FC mit Parametern vom Datentyp ANY übergeben werden können.



In diesem Beispiel hat FC100 drei Parameter (*in_par1*, *in_par2* und *in_par3*), die als Datentyp ANY deklariert wurden.

- Wenn FB10 die FC100 aufruft, übergibt FB10 eine Ganzzahl (die statische Variable Drehzahl), ein Wort (MW100) und ein Doppelwort in DB10 (DB10.DBD40).
- Wenn FB11 die FC10 aufruft, übergibt FB11 ein Feld aus Realzahlen (die temporäre Variable "Thermo"), einen booleschen Wert (M 1.3) und eine Zeit (T2).

Angeben eines Datenbereichs für einen Parameter ANY

Sie können einem Parameter ANY aber nicht nur individuelle Operanden (z. B. MW100) zuordnen, sondern Sie können auch einen Datenbereich angeben. Wenn Sie einen Datenbereich als Aktualparameter zuordnen möchten, verwenden Sie das folgende Format einer Konstanten, um die Datenmenge anzugeben, die übergeben werden soll:

p# *Bereichskennung Byte.Bit Datentyp Wiederholfaktor*

Sie können für das Element *Datentyp* in dem Format für Konstanten alle elementaren Datentypen und den Datentyp DATE_AND_TIME angeben. Handelt es sich bei dem Datentyp nicht um BOOL, dann muß die Bitadresse von 0 (x.0) angegeben werden. Nachfolgende Tabelle zeigt Beispiele für das Format zum Angeben von Speicherbereichen, die an einen Parameter ANY übergeben werden sollen.

Aktualparameter	Beschreibung
p# M 50.0 BYTE 10	Gibt 10 Bytes im Speicherbereich Merker an: MB50 bis MB59.
p# DB10.DBX5.0 S5TIME 3	Gibt 3 Einheiten von Daten vom Datentyp S5TIME an, die in DB10 gespeichert sind: DB Byte 5 bis DB Byte 10.
p# A 10.0 BOOL 4	Gibt 4 Bits im Speicherbereich Ausgänge an: A 10.0 bis A 10.3.

Beispiel für die Verwendung des Parametertyps ANY

Das folgende Beispiel zeigt, wie Sie mit Hilfe des Parametertyps ANY und der Systemfunktion SFC 20 BLKMOV einen Speicherbereich von 10 Byte kopieren können.

```
AWL                     Erläuterung
FUNCTION FC 10:VOID
VAR_TEMP
  Quelle  : ANY;
  Ziel    : ANY;
END_VAR
BEGIN
LAR1    P#Quelle;       Lade die Anfangsadresse des ANY-Pointers in AR1.

L  B#16#10;             Lade die Syntax-ID und
T  LB[AR1,P#0.0];       transferiere sie in den ANY-Pointer.

L  B#16#02;             Lade Datentyp Byte und
T  LB[AR1,P#1.0];       Transferiere ihn in den ANY-Pointer.

L  10;                  Lade 10 Byte und
T  LW[AR1,P#2.0];       transferiere sie in den ANY-Pointer.

L  22;                  Quelle ist der DB22, DBB11
T  LW[AR1,P#4.0];
L  P#DBX11.0;
T  LD[AR1,P#6.0];

LAR1    P#Ziel;         Lade die Anfangsadresse des ANY-Pointers in AR1.

L  B#16#10;             Lade die Syntax-ID und
T  LB[AR1,P#0.0];       transferiere sie in den ANY-Pointer.

L  B#16#02;             Lade Datentyp Byte und
T  LB[AR1,P#1.0];       Transferiere ihn in den ANY-Pointer.

L  10;                  Lade 10 Byte und
T  LW[AR1,P#2.0];       transferiere sie in den ANY-Pointer.

L  33;                  Ziel ist der DB33, DBB202
T  LW[AR1,P#4.0];
L  P#DBX202.0;
T  LD[AR1,P#6.0];

CALL SFC 20 (           Aufruf der Systemfunktion Blockmove
    SRC BLK := Quelle,
    RET_VAL := MW 12,   Auswertung des BIE-Bit und des MW 12
    DSTBLK := Ziel
     );
END FUNCTION
```

A.3.4.7 Zuordnen von Datentypen zu Lokaldaten von Codebausteinen

STEP 7 begrenzt die Datentypen (elementare und zusammengesetzte Datentypen sowie Parametertypen), die in der Variablendeklaration den Lokaldaten eines Bausteins zugeordnet werden können.

Gültige Datentypen für die Lokaldaten eines OBs

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für einen OB. Weil Sie einen OB nicht aufrufen können, kann ein OB auch nicht über Parameter (Ein-, Aus- oder Durchgänge) verfügen. Weil ein OB keinen Instanz-DB hat, können Sie für einen OB keine statischen Variablen deklarieren. Bei den Datentypen der temporären Variablen eines OB kann es sich um elementare oder zusammengesetzte Datentypen sowie um ANY handeln.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarations-typ	Elementare Datentypen	Zusammen-gesetzte Datentypen	Para-metertyp	Para-metertyp	Para-metertyp	Para-metertyp	Para-metertyp
			TIMER	**COUNTER**	**BLOCK**	**POINTER**	**ANY**
Eingang	—	—	—	—	—	—	—
Ausgang	—	—	—	—	—	—	—
Durchgang	—	—	—	—	—	—	—
Statisch	—	—	—	—	—	—	—
Temporär	●(1)	●(1)	—	—	—	—	●(1)

(1) Im L-Stack des OB gespeichert.

Gültige Datentypen für die Lokaldaten eines FBs

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für einen FB. Wegen des Instanz-DB gibt es weniger Einschränkungen beim Deklarieren von Lokaldaten für einen FB. Für die Deklaration von Eingangsparametern gibt es keinerlei Einschränkungen, für einen Ausgangsparameter können Sie keine Parametertypen deklarieren und für Durchgangsparameter sind nur die Parametertypen POINTER und ANY zulässig. Temporäre Variablen können Sie als Datentyp ANY deklarieren. Alle anderen Parametertypen sind unzulässig.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	●	●	●	●	●	●
Ausgang	●	●	—	—	—	—	—
Durchgang	●	●(1)(3)	—	—	—	●	●
Statisch	●	●	—	—	—	—	—
Temporär	●(2)	●(2)	—	—	—	—	●(2)

(1) Als Verweis (48-Bit-Pointer) im Instanz-DB gespeichert.

(2) Im L-Stack des FB gespeichert.

(3) STRINGS können nur in der Standardlänge definiert werden.

Gültige Datentypen für die Lokaldaten eines FCs

Nachfolgende Tabelle zeigt die Einschränkungen (–) beim Deklarieren von Lokaldaten für eine FC. Weil eine FC keinen Instanz-DB hat, verfügt sie auch nicht über statische Variablen. Für Eingangs-, Ausgangs- und Durchgangsparameter einer FC sind die Parametertypen POINTER und ANY zulässig. Sie können auch temporäre Variablen vom Parametertyp ANY deklarieren.

Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

Deklarationstyp	Elementare Datentypen	Zusammengesetzte Datentypen	Parametertyp	Parametertyp	Parametertyp	Parametertyp	Parametertyp
			TIMER	COUNTER	BLOCK	POINTER	ANY
Eingang	●	●(2)	●	●	●	●	●
Ausgang	●	●(2)	—	—	—	●	●
Durchgang	●	●(2)	—	—	—	●	●
Temporär	●(1)	●(1)	—	—	—	—	●(1)

(1) Im L-Stack der FC gespeichert.

(2) STRINGS können nur in der Standardlänge definiert werden.

A.3.4.8 Zulässige Datentypen beim Übergeben von Parametern

Regeln für die Parameterübergabe zwischen Bausteinen

Wenn Sie Formalparametern Aktualparameter zuordnen, dann können Sie entweder eine absolute Adresse, einen symbolischen Namen oder eine Konstante angeben. STEP 7 begrenzt die jeweils gültigen Zuordnungen bei den verschiedenen Parametern. Ausgangs- und Durchgangsparameter können zum Beispiel keinem konstanten Wert zugeordnet werden (weil der Zweck eines Ausgangs bzw. eines Durchgangs das Ändern des Werts ist). Diese Einschränkungen gelten besonders bei Parametern mit zusammengesetzten Datentypen, denen weder eine absolute Adresse noch eine Konstante zugeordnet werden kann.

Nachfolgende Tabellen zeigen die Einschränkungen (–) bei den Datentypen von Aktualparametern, die Formalparametern zugeordnet werden.

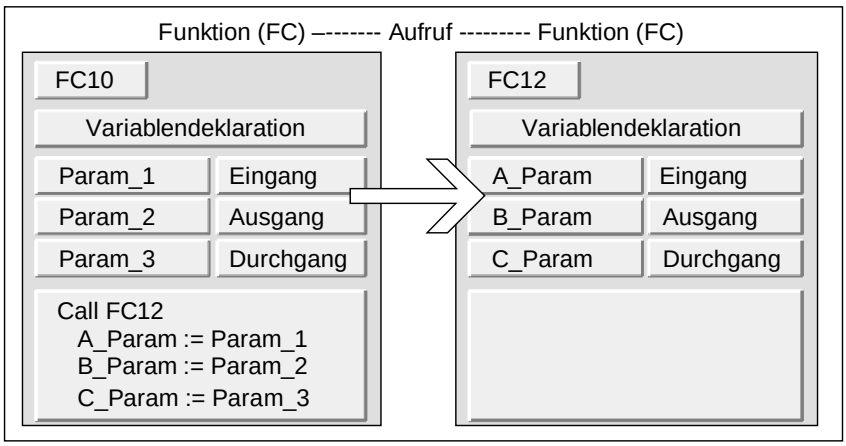
Die gültigen Zuordnungen sind durch das Zeichen ● markiert.

		Elementare Datentypen		
Deklarationstyp	**Absolute Adresse**	**Symbolischer Name (in der Symboltabelle)**	**Bausteinlokales Symbol**	**Konstante**
Eingang	●	●	●	●
Ausgang	●	●	●	—
Durchgang	●	●	●	—

Zusammengesetzte Datentypen				
Deklarationstyp	**Absolute Adresse**	**Symbolischer Name des Elements des DB (in der Symboltabelle)**	**Bausteinlokales Symbol**	**Konstante**
Eingang	—	●	●	—
Ausgang	—	●	●	—
Durchgang	—	●	●	—

Zulässige Datentypen beim Aufruf einer FC durch eine andere FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern einer aufgerufenen FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.



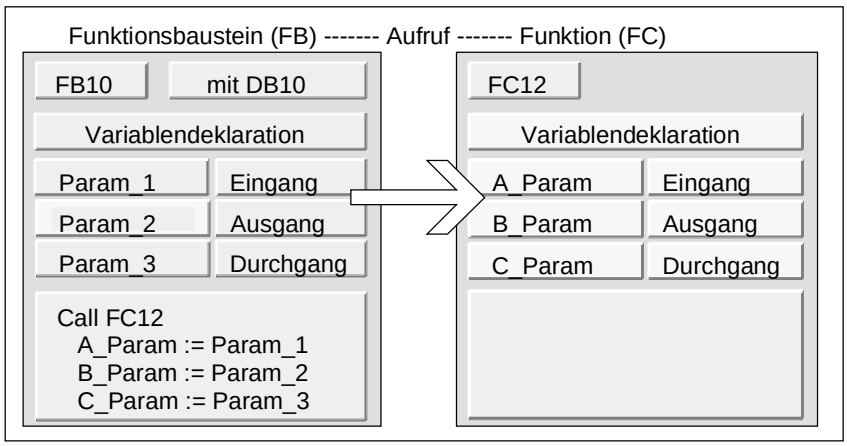
STEP 7 begrenzt die Zuordnung von Formalparametern einer FC als Aktualparameter zu den Formalparametern einer anderen FC. Sie können beispielsweise keine Parameter mit zusammengesetztem Datentyp oder Parametertyp als Aktualparameter zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC eine andere FC aufruft.

Deklarationstyp	**Elementare Datentypen**	**Zusammengesetzte Datentypen**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**
			TIMER	**COUNTER**	**BLOCK**	**POINTER**	**ANY**
Eingang → Eingang	●	—	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf einer FC durch einen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern einer aufrufenden FC zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FC 12 zugeordnet werden.

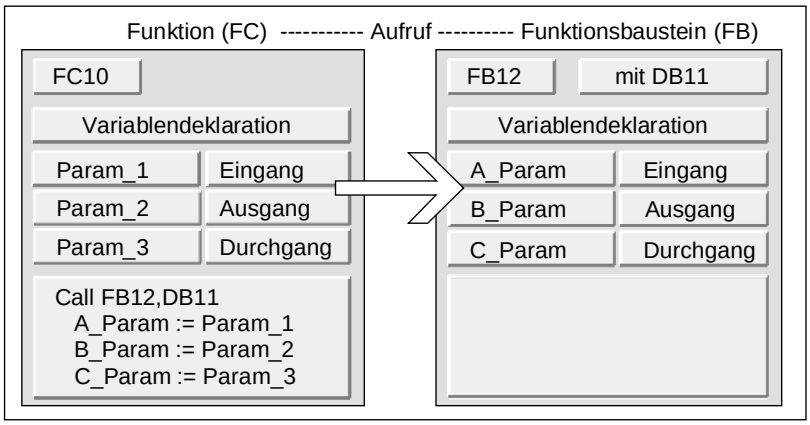


STEP 7 begrenzt die Zuordnung der Formalparameter einer FB zu den Formalparametern einer FC. Sie können beispielsweise keine Parameter mit Parametertyp als Aktualparameter zuordnen. Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB eine FC aufruft.

Deklarationstyp	**Elementare Datentypen**	**Zusammengesetzte Datentypen**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**
			TIMER	**COUNTER**	**BLOCK**	**POINTER**	**ANY**
Eingang → Eingang	●	●	—	—	—	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	●	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch eine FC

Sie können die Formalparameter einer aufrufenden FC den Formalparametern eines aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FC 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



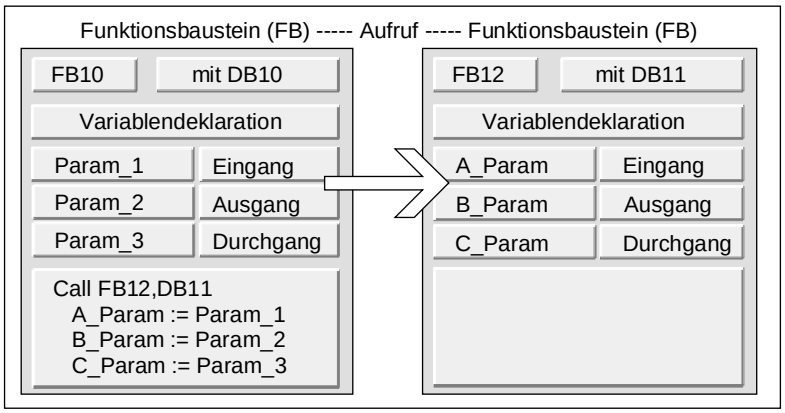
STEP 7 begrenzt die Zuordnung von Formalparametern einer FC zu den Formalparametern eines FB. Sie können beispielsweise Parameter mit zusammengesetztem Datentyp nicht als Aktualparameter zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn eine FC einen FB aufruft.

Deklarationstyp	**Elementare Datentypen**	**Zusammengesetzte Datentypen**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**
			TIMER	**COUNTER**	**BLOCK**	**POINTER**	**ANY**
Eingang → Eingang	●	—	●	●	●	—	—
Eingang → Ausgang	—	—	—	—	—	—	—
Eingang → Durchgang	—	—	—	—	—	—	—
Ausgang → Eingang	—	—	—	—	—	—	—
Ausgang → Ausgang	●	—	—	—	—	—	—
Ausgang → Durchgang	—	—	—	—	—	—	—
Durchgang → Eingang	●	—	—	—	—	—	—
Durchgang → Ausgang	●	—	—	—	—	—	—
Durchgang → Durchgang	●	—	—	—	—	—	—

Zulässige Datentypen beim Aufruf eines FB durch einen anderen FB

Sie können die Formalparameter eines aufrufenden FB den Formalparametern des aufgerufenen FB zuordnen. Nachfolgendes Bild zeigt die Formalparameter von FB 10, die als Aktualparameter den Formalparametern von FB 12 zugeordnet werden.



STEP 7 begrenzt die Zuordnung von Formalparametern eines FB zu den Formalparametern eines anderen FB. Sie können beispielsweise keine Eingangs- und Ausgangsparameter mit zusammengesetztem Datentyp als Aktualparameter den Eingangs- und Ausgangsparametern eines aufgerufenen FB zuordnen. Sie können aber Eingangsparameter mit dem Parametertyp TIMER, COUNTER oder BLOCK den Eingangsparametern des aufgerufenen FB zuordnen.

Nachfolgende Tabelle zeigt die zulässigen Datentypen (●), wenn ein FB einen anderen FB aufruft.

Deklarationstyp	**Elementare Datentypen**	**Zusammengesetzte Datentypen**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**	**Parametertyp**
			TIMER	**COUNTER**	**BLOCK**	**POINTER**	**ANY**
Eingang→ Eingang	●	●	●	●	●	—	—
Eingang→ Ausgang	—	—	—	—	—	—	—
Eingang→ Durchgang	—	—	—	—	—	—	—
Ausgang→ Eingang	—	—	—	—	—	—	—
Ausgang→ Ausgang	●	●	—	—	—	—	—
Ausgang→ Durchgang	—	—	—	—	—	—	—
Durchgang→ Eingang	●	—	—	—	—	—	—
Durchgang→ Ausgang	●	—	—	—	—	—	—
Durchgang→ Durchgang	●	—	—	—	—	—	—

A.3.4.9 Übergabe an IN_OUT-Parameter eines FB

Bei der Übergabe von zusammengesetzten Datentypen an IN_OUT-Parameter eines Funktionsbausteins (FB) wird die Operandenadresse der Variable übergeben (call by reference).

Bei der Übergabe von elementaren Datentypen an IN_OUT-Parameter eines FB werden die Werte in den Instanz-Datenbaustein kopiert, bevor der FB gestartet wird und aus dem Instanz-Datenbaustein herauskopiert, nachdem der FB beendet ist.

Dadurch können IN_OUT-Variablen elementaren Datentyps mit einem Wert initialisiert werden.

Es ist jedoch nicht möglich, in einem Aufruf eine Konstante als Aktualparameter an der Position einer IN_OUT-Variablen anzugeben, da auf eine Konstante nicht geschrieben werden kann.

Variable vom Datentyp STRUCT oder ARRAY können nicht initialisiert werden, da in diesem Fall im Instanz-Datenbaustein nur eine Adresse liegt.

A.4 Arbeiten mit älteren Projekten

A.4.1 Umsetzen von Version 1-Projekten

Sie haben die Möglichkeit, Projekte wiederzuverwenden, die mit STEP 7 Version 1 erstellt wurden. Dazu wandeln Sie Projekte der Version 1 in Projekte der Version 2 um.

Folgende Komponenten eines Version 1-Projekts bleiben dabei erhalten:

- Projektstruktur mit Programmen
- Bausteine
- AWL-Quelldateien
- Symboltabelle

Die Konfiguration der Hardware wird nicht konvertiert. Die erhaltenen Programmkomponenten können Sie in andere Projekte kopieren. Sie können auch das neue Projekt um eine Station ergänzen, die Sie entsprechend konfigurieren und parametrieren.

Nach dem Umsetzen in Version 2 können Sie in einem Dialog entscheiden, ob nun dieses Version 2-Projekt in ein Projekt Ihrer aktuellen STEP 7-Version umgesetzt werden soll.

Hinweis

Die einzelnen Bausteine bleiben in ihren Eigenschaften Bausteine der Version 1. Der in Version 1 generierte Code wird nicht verändert, und die Bausteine lassen sich daher nicht im Zusammenhang mit Multiinstanzen verwenden.

Wenn Sie in den umgesetzten Bausteinen Multiinstanzen vereinbaren wollen, generieren Sie mit der Applikation "KOP/AWL: Bausteine programmieren" zunächst aus den umgesetzten Bausteinen AWL-Quellen und übersetzen diese wieder in Bausteine.

Das Programmieren von Multiinstanzen ist eine neue Möglichkeit in STEP 7 Version 2 für die Erstellung von Funktionsbausteinen (FBs). Wenn Sie die mit Version 1 erstellten FBs in der gleichen Funktionalität im Version 2-Projekt weiterverwenden wollen, brauchen Sie diese nicht umsetzen.

Vorgehensweise

Um Version 1-Projekte umzusetzen gehen Sie wie folgt vor:

1. Wählen Sie den Menübefehl **Datei > Version 1-Projekt öffnen**.
2. Wählen Sie im folgenden Dialogfeld das Version 1-Projekt aus, das Sie wiederverwenden möchten. Ein Version 1-Projekt erkennen Sie an der Dateierweiterung *.s7a.
3. Geben Sie im nachfolgenden Dialogfeld den Namen des neuen Projekts an, in welches das Version 1-Projekt umgesetzt werden soll.

A.4.2 Umsetzen von Version 2-Projekten

In STEP 7 können Sie mit dem Menübefehl **Datei > Öffnen** auch Version 2-Projekte öffnen.

Version 2-Projekte/Bibliotheken lassen sich mit dem Menübefehl **Datei > Speichern unter** und der Einstellung "mit Reorganisation" auf Ihre aktuelle STEP 7-Version umsetzen (Migration). Das Projekt wird dann als Projekt der aktuellen STEP 7-Version gespeichert.

Sie können Projekte und Bibliotheken aus älteren STEP 7-Version unter Beibehaltung des Formats bearbeiten und speichern, in dem Sie im Dialogfeld **"Projekt speichern unter"** den Dateityp der älteren STEP 7-Version auswählen. Um beispielsweise die Objekte mit STEP 7 Version 2.1 zu bearbeiten, wählen Sie hier Projekt 2.x bzw. Bibliothek 2.x (als Version 2 speichern ist nicht mehr möglich ab Version 5.1, siehe auch Version 2-Projekte und Bibliotheken bearbeiten).

Bezeichnungen des Dateityps

	STEP 7 V3	**ab STEP 7 V4**
Dateityp der aktuellen Version	Projekt3.x Bibliothek3.x	Projekt Bibliothek
Dateityp der Vorgängerversion	Projekt2.x Bibliothek2.x	Projekt2.x Bibliothek2.x

Dann steht Ihnen allerdings nur der Funktionsumfang der älteren STEP 7-Version zur Verfügung. Die Projekte bzw. Bibliotheken können danach aber weiterhin mit der älteren STEP 7-Version gepflegt werden.

Hinweis

Bei Übergang von Version 3 nach Versionen ab 4 wurde lediglich die Bezeichnung geändert, das Format blieb identisch. Deshalb gibt es in STEP 7 V4 keinen Dateityp Projekt3.x.

Vorgehensweise

Um Version 2-Projekte im Format der aktuellen STEP 7-Version umzusetzen, gehen Sie wie folgt vor:

1. Führen Sie für das Projekt die Funktion "Speichern unter (Menü Datei) " mit Reorganisation aus."
2. Wählen Sie im Dialogfeld "Projekt speichern unter" den Dateityp "Projekt" und betätigen Sie die Schaltfläche "Speichern".

Um Version 2-Projekte unter Beibehaltung des Formats in die aktuelle STEP 7-Version umzusetzen, gehen Sie wie folgt vor:

1. Führen Sie bei Bedarf Schritt 1 wie oben beschrieben durch.
2. Wählen Sie im Dialogfeld "Projekt speichern unter" den Dateityp der älteren STEP 7-Version und betätigen Sie die Schaltfläche "Speichern".

A.4.3 Hinweise zu STEP 7 V2.1-Projekten mit GD-Kommunikation

- Wenn Sie ein Projekt mit Globaldaten aus STEP 7 V2.1 nach STEP 7 V5 umsetzen wollen, müssen Sie zuvor die GD-Tabelle mit STEP 7 V5.0 im STEP 7 V2.1 Projekt öffnen. Damit werden Ihre bisher projektierten Kommunikationsdaten durch GD-Kommunikation automatisch in die neue Struktur umgewandelt.
- Beim Archivieren von STEP 7 V2.1-Projekten kann es zu einer Fehlermeldung älterer Packprogramme (ARJ, PKZIP...) kommen, wenn im Projekt Dateien mit Namen größer acht Zeichen vorhanden sind. Diese Meldung erscheint auch, wenn in dem STEP 7 V2.1-Projekt das MPI-Netz mit einer Bezeichnung größer 8 Zeichen editiert wurde. Editieren Sie in STEP 7 V2.1-Projekten mit Globaldaten für das MPI-Netz einen Namen mit maximal 8 Zeichen Länge, bevor Sie die Konfigurierung der Globaldaten-Kommunikation erstmals starten.
- Wollen Sie ein Projekt aus STEP 7 V2.1 umbenennen, so müssen Sie die Überschriften der Spalten (CPU's) in der GD-Tabelle neu zuordnen, indem Sie die zugehörige CPU neu auswählen. Wenn Sie den alten Projektnamen wiederherstellen, sind die Zuordnungen wieder vorhanden.

A.4.4 Erweitern von DP-Slaves, die mit Vorgänger-Versionen von STEP 7 erstellt wurden

Konstellationen, die durch das Einspielen neuer GSD-Dateien entstehen können

Neue DP-Slaves können durch die Installation neuer GSD-Dateien in den Hardware-Katalog von HW Konfig aufgenommen werden. Nach der Installation sind sie im Ordner "Weitere Feldgeräte" verfügbar.

Sie können einen modularen DP-Slave nicht mehr mit STEP 7, Servicepack 3, wie gewohnt umkonfigurieren bzw. erweitern, wenn er

- mit einer Vorgänger-Version von STEP 7 konfiguriert wurde und
- im Hardware Katalog nicht durch eine GSD-, sondern durch eine Typdatei repräsentiert wurde und
- durch eine neue GSD-Datei überinstalliert wurde

Abhilfe

Wenn Sie den DP-Slave mit **neuen Baugruppen**, die in der GSD-Datei beschrieben sind, einsetzen wollen:

- Löschen Sie den DP-Slave und projektieren Sie ihn neu – der DP-Slave wird dann nicht durch die Typ-, sondern komplett durch die GSD-Datei beschrieben.

Wenn Sie **keine neuen Baugruppen**, die nur in der GSD-Datei beschrieben sind, einsetzen wollen:

- Wählen Sie den Ordner "Weitere Feldgeräte/Kompatible Profibus-DP-Slaves" unter PROFIBUS-DP im Fenster "Hardware Katalog". In diesen Ordner verschiebt STEP 7 die "alten" Typdateien, wenn sie durch neue GSD-Dateien

ersetzt werden. In diesem Ordner finden Sie die Baugruppen, mit denen Sie den bereits konfigurierten DP-Slave erweitern können.

A.4.5 DP-Slaves mit fehlenden oder fehlerhaften GSD-Dateien

Wenn Sie ältere Stationskonfigurationen mit STEP 7 Version 5.1 bearbeiten, dann kann es in seltenen Fällen vorkommen, daß die GSD-Datei eines DP-Slaves fehlt oder nicht übersetzbar ist (z. B. wegen Syntaxfehlern in der GSD-Datei).

In diesem Fall erzeugt STEP 7 einen "Dummy"-Slave, der den projektierten Slave repräsentiert, z. B. nach einem Stations-Laden in PG oder nach dem Öffnen und Weiterbearbeiten eines älteren Projekts. Dieser "Dummy"-Slave ist nur sehr eingeschränkt bearbeitbar; Sie können den Slaveaufbau (DP-Kennungen) und die Slaveparameter nicht ändern. Ein erneutes Laden in die Station ist aber möglich, die ursprüngliche Konfiguration des Slaves bleibt erhalten. Ebenso kann der gesamte DP-Slave gelöscht werden.

Umkonfigurieren und -parametrieren des DP-Slaves

Wenn Sie den DP-Slave umkonfigurieren oder umparametrieren möchten, dann müssen Sie eine aktuelle GSD-Datei für diesen DP-Slave vom Hersteller anfordern und über den Menübefehl **Extras > Neue GSD installieren** verfügbar machen.

Nachdem die korrekte GSD-Datei installiert ist, wird sie zur Darstellung des DP-Slaves benutzt. Der DP-Slave behält seine Daten und ist wieder voll bearbeitbar.

A.5 Beispielprogramme

A.5.1 Beispielprojekte und Beispielprogramme

Die Installations-CD enthält eine Anzahl von Beispielprojekten. Bei den Projekten, die nicht in diesem Kapitel beschrieben werden, ist eine Beschreibung im entsprechenden OB 1 enthalten.

Beispiele und Beispielprojekte	**Auf CD enthalten**	**In diesem Kapitel beschrieben**	**Beschreibung dt./engl. in OB 1**
Projekt "ZDt01_08_STEP7_Mischen" (Industrieller Mischprozeß)	•	•	
Projekte "ZDt01_01_STEP7_*" bis "ZDt01_06_STEP7_* (Erste Schritte und Übungen)	•	Eigenes Handbuch	•
Projekt ""ZDt01_09_STEP7_Zebra" (Ampelsteuerung am Zebrastreifen)	•		•
Projekt ""ZDt01_10_STEP7_COM_SFB" (Datenaustausch zwischen zwei S7-400-CPUs)	•		•
Projekt ""ZDt01_11_STEP7_COM_SFC1" und ""ZDt01_12_STEP7_COM_SFC2" (Datenaustausch über Kommunikations-SFCs für nichtprojektierte Verbindungen)	•		•
Beispiel zur Hantierung mit Uhrzeitalarmen		•	
Beispiel zur Hantierung mit Verzögerungsalarmen		•	
Beispiel zum Maskieren und Demaskieren von Synchronfehlerereignissen		•	
Beispiel zum Sperren und Freigeben von Alarm- und Asynchronereignissen		•	
Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronereignissen		•	

Der Schwerpunkt der Beispiele liegt nicht auf der Vermittlung eines bestimmten Programmierstils oder technischen Fachwissens, das Sie zur Steuerung eines bestimmten Prozesses benötigen. Mit Hilfe dieser Beispiele können Sie vielmehr die Schritte nachvollziehen, die beim Programmentwurf ausgeführt werden müssen.

Löschen und Installieren von mitgelieferten Beispielprojekten

Die mitgelieferten Beispielprojekte können Sie im SIMATIC Manager löschen und danach wieder neu installieren. Für die Installation starten Sie das Programm Setup von STEP 7 V5.0. Die Beispielprojekte lassen sich selektiv nachinstallieren.

Hinweis

Die mitgelieferten Beispielprojekte werden, falls diese nicht abgewählt sind, bei einer STEP 7-Installation kopiert. Wenn Sie mitgelieferte Beispielprojekte bearbeitet haben, werden bei einer erneuten STEP 7- Installation die von Ihnen geänderten Beispielprojekte mit dem Original überschrieben.

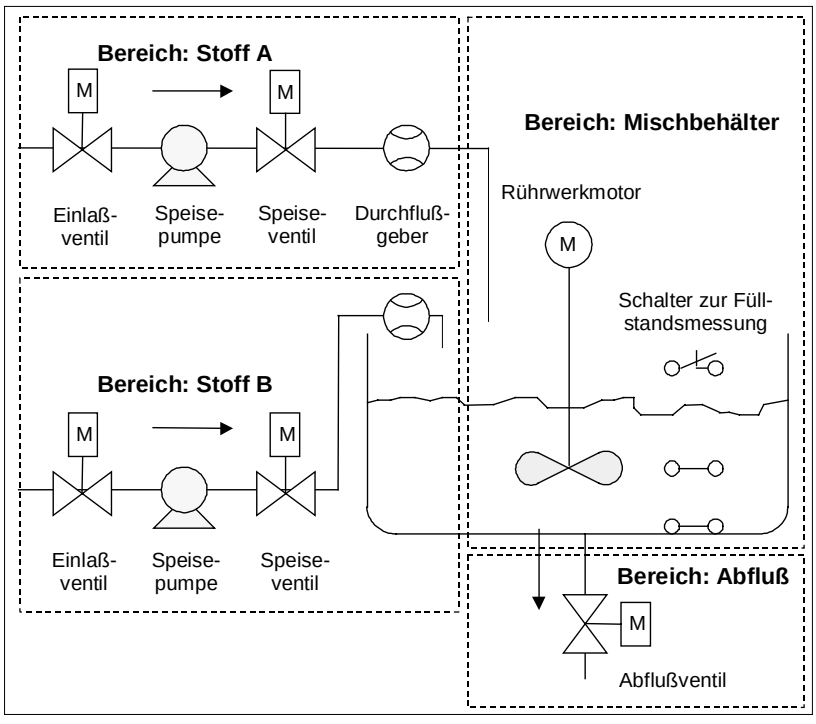
Deswegen sollten Sie mitgelieferte Beispielprojekte vor Änderungen kopieren und nur die Kopie bearbeiten.

A.5.2 Beispielprogramm für einen industriellen Mischprozeß

Das Beispielprogramm baut auf den Informationen auf, die bereits im Teil 1 des Handbuchs über das Steuern eines industriellen Mischprozesses vermittelt wurden.

Aufgabenstellung

Zwei Stoffe (Stoff A und Stoff B) sollen in einem Mischbehälter durch ein Rührwerk vermengt werden. Die Masse soll durch ein Abflußventil aus dem Mischbehälter abgelassen werden. Im nachfolgenden Bild sehen Sie ein Diagramm des Beispielprozesses.



Beschreibung der Teilprozesse

Im Teil 1 des Handbuchs wurde beschrieben, wie Sie den Beispielprozeß in Funktionsbereiche und einzelne Aufgaben zerlegen können. Im folgenden finden Sie eine Beschreibung der einzelnen Teilbereiche.

Bereiche Stoff A und Stoff B:

- Die Stoffzuleitungen sind jeweils mit einem Einlaß- und einem Speiseventil sowie einer Speisepumpe zu versehen.
- In den Zuleitungen befinden sich Durchflußgeber.
- Das Einschalten der Speisepumpen muß verriegelt sein, wenn der Füllstandsmesser ”Behälter ist voll” anzeigt.
- Das Einschalten der Speisepumpen muß verriegelt sein, wenn das Abflußventil geöffnet ist.
- Frühestens 1 Sekunde nach Ansteuerung der Speisepumpe dürfen Einlaß- und Speiseventil geöffnet werden.
- Sofort nach dem Stoppen der Speisepumpen (Signal des Durchflußgebers) müssen die Ventile geschlossen werden, um ein Ablaufen des Stoffes aus der Pumpe zu verhindern.
- Die Ansteuerung der Speisepumpen ist zeitüberwacht, d. h. innerhalb von 7 Sekunden nach Ansteuerung muß vom Durchflußgeber ein Durchfluß gemeldet werden.
- Die Speisepumpen müssen in kürzester Zeit abgeschaltet werden, wenn die Durchflußgeber während des Laufs der Speisepumpen keinen Durchfluß mehr melden.
- Die Anzahl der Starts der Speisepumpen müssen gezählt werden (Wartungsintervall).

Bereich Mischbehälter:

- Das Einschalten des Rührwerkmotors muß verriegelt sein, wenn der Füllstandsmesser ”Behälter unter Minimum” anzeigt oder das Abflußventil offen ist.
- Der Rührwerkmotor gibt ein Rückmeldesignal nach Erreichen der Nenndrehzahl. Wird dieses Signal nicht innerhalb von 10 Sekunden nach Ansteuerung des Motors gemeldet, muß der Motor abgeschaltet werden.
- Die Anzahl der Starts des Rührwerkmotors müssen gezählt werden (Wartungsintervall).
- Im Mischbehälter sind drei Geber zu installieren:
  - Behälter voll: Öffner. Ist der maximale Füllstand erreicht, wird der Kontakt geöffnet.
  - Füllstand im Behälter über Minimum: Schließer. Ist der minimale Füllstand erreicht, wird der Kontakt geschlossen.
  - Behälter nicht leer: Schließer. Ist der Behälter nicht leer, ist der Kontakt geschlossen.

Bereich Abfluß:

- Der Abfluß soll über ein Magnetventil gesteuert werden.
- Das Magnetventil wird durch den Bediener gesteuert, jedoch spätestens beim Signal ”Behälter leer” wieder geschlossen.
- Das Öffnen des Abflußventils ist verriegelt, wenn
  - der Rührwerkmotor läuft
  - der “Behälter leer” ist

Bedienpult

Um einem Bediener das Starten und Stoppen sowie das Beobachten des Prozesses zu ermöglichen, soll außerdem ein Bedienpult angelegt werden. Auf dem Bedienpult befinden sich

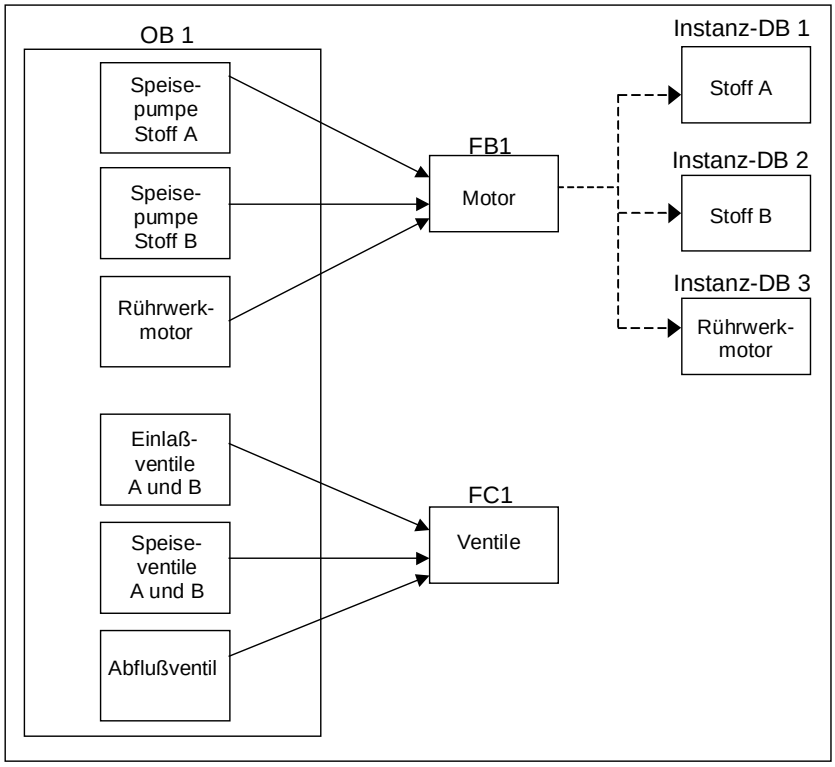
- Taster zum Steuern der wichtigsten Vorgänge. Mit dem Taster “Wartungsanzeige rücksetzen” schalten Sie die Wartungsanzeigelampen derjenigen Motoren aus, bei denen eine Wartung fällig ist, und setzen die zugehörigen Zählerstände für das Wartungsintervall auf 0.
- Anzeigelampen zum Ablesen des Betriebszustands.
- der NOT-AUS-Schalter.

A.5.2.1 Definieren von Codebausteinen

Mit dem Verteilen des Anwenderprogramms auf verschiedene Bausteine und der Hierarchie der Bausteinaufrufe legen Sie die Struktur des Programms fest.

Hierarchie der Bausteinaufrufe

Nachfolgendes Bild zeigt die Hierarchie der Bausteine, die im strukturierten Programm aufgerufen werden sollen.



- OB1: Er ist die Schnittstelle zum Betriebssystem der CPU und enthält das Hauptprogramm. Im OB 1 werden die Bausteine FB 1 und FC 1 aufgerufen und die spezifischen Parameter übergeben, die für die Steuerung des Prozesses erforderlich sind.
- FB 1: Die Speisepumpe für Stoff A, die Speisepumpe für Stoff B und der Rührwerkmotor können durch einen einzigen Funktionsbaustein gesteuert werden, da die Anforderungen (Ein-, Ausschalten, Einsätze zählen, etc.) identisch sind.
- Instanz-DB 1-3: Die Aktualparameter und die statischen Daten für die Steuerung der Speisepumpen für Stoff A, Stoff B und für den Rührwerkmotor sind unterschiedlich und werden deshalb in drei dem FB 1 zugeordneten Instanz-DBs abgelegt.
- FC 1: Die Einlaß- und Speiseventile für die Stoffe A und B sowie das Abflußventil verwenden ebenfalls einen gemeinsamen Codebaustein. Da nur die Funktion Öffnen und Schließen programmiert werden muß, genügt ein einziger FC.

A.5.2.2 Zuordnen von symbolischen Namen

Definieren symbolischer Namen

In dem Beispielprogramm werden Symbole verwendet, die Sie mit STEP 7 in der Symboltabelle definieren müssen. Nachfolgende Tabellen zeigen die symbolischen Namen und die absoluten Adressen der verwendeten Programmelemente.

Symbolische Adressen von Speisepumpe, Rührwerkmotor und Einlaßventile			
Symbolischer Name	**Adresse**	**Datentyp**	**Beschreibung**
Feed_pump_A_start	E 0.0	BOOL	Starttaster Speisepumpe für Stoff A
Feed_pump_A_stop	E 0.1	BOOL	Stopptaster Speisepumpe für Stoff A
Flow_A	E 0.2	BOOL	Stoff A fließt
Inlet_valve_A	A 4.0	BOOL	Ansteuerung des Einlaßventils für Stoff A
Feed_valve_A	A 4.1	BOOL	Ansteuerung des Speiseventils für Stoff A
Feed_pump_A_on	A 4.2	BOOL	Anzeigelampe “Speisepumpe Stoff A läuft”
Feed_pump_A_off	A 4.3	BOOL	Anzeigelampe “Speisepumpe Stoff A läuft nicht”
Feed_pump_A	A 4.4	BOOL	Ansteuerung der Speisepumpe für Stoff A
Feed_pump_A_fault	A 4.5	BOOL	Anzeigelampe “Fehler der Speisepumpe A”
Feed_pump_A_maint	A 4.6	BOOL	Anzeigelampe “Wartung der Speisepumpe A erforderlich”
Feed_pump_B_start	E 0.3	BOOL	Starttaster Speisepumpe für Stoff B
Feed_pump_B_stop	E 0.4	BOOL	Stopptaster Speisepumpe für Stoff B
Flow_B	E 0.5	BOOL	Stoff B fließt
Inlet_valve_B	A 5.0	BOOL	Ansteuerung des Einlaßventils für Stoff B
Feed_valve_B	A 5.1	BOOL	Ansteuerung des Speiseventils für Stoff B
Feed_pump_B_on	A 5.2	BOOL	Anzeigelampe “Speisepumpe Stoff B läuft”
Feed_pump_B_off	A 5.3	BOOL	Anzeigelampe “Speisepumpe Stoff B läuft nicht”
Feed_pump_B	A 5.4	BOOL	Ansteuerung der Speisepumpe für Stoff B
Feed_pump_B_fault	A 5.5	BOOL	Anzeigelampe “Fehler der Speisepumpe B”
Feed_pump_B_maint	A 5.6	BOOL	Anzeigelampe “Wartung der Speisepumpe B erforderlich”
Agitator_running	E 1.0	BOOL	Rückmeldesignal des Rührwerkmotors
Agitator_start	E 1.1	BOOL	Starttaster Rührwerk
Agitator_stop	E 1.2	BOOL	Stopptaster Rührwerk

Symbolische Adressen von Speisepumpe, Rührwerkmotor und Einlaßventile			
Symbolischer Name	**Adresse**	**Datentyp**	**Beschreibung**
Agitator	A 8.0	BOOL	Ansteuerung des Rührwerks
Agitator_on	A 8.1	BOOL	Anzeigelampe “Rührwerk läuft”
Agitator_off	A 8.2	BOOL	Anzeigelampe “Rührwerk läuft nicht”
Agitator_fault	A 8.3	BOOL	Anzeigelampe “Fehler des Rührwerkmotors”
Agitator_maint	A 8.4	BOOL	Anzeigelampe “Wartung des Rührwerkmotors erforderlich”

Symbolische Adressen von Sensoren und Behälterfüllstandsanzeige			
Symbolischer Name	**Adresse**	**Datentyp**	**Beschreibung**
Tank_below_max	E 1.3	BOOL	Sensor “Mischbehälter nicht voll”
Tank_above_min	E 1.4	BOOL	Sensor “Mischbehälter über Minimum”
Tank_not_empty	E 1.5	BOOL	Sensor “Mischbehälter nicht leer”
Tank_max_disp	A 9.0	BOOL	Anzeigelampe “Mischbehälter voll”
Tank_min_disp	A 9.1	BOOL	Anzeigelampe “Mischbehälter unter Minimum”
Tank_empty_disp	A 9.2	BOOL	Anzeigelampe “Mischbehälter leer”

Symbolische Adressen des Abflußventils			
Symbolischer Name	**Adresse**	**Datentyp**	**Beschreibung**
Drain_open	E 0.6	BOOL	Taster zum Öffnen des Abflußventils
Drain_closed	E 0.7	BOOL	Taster zum Schließen des Abflußventils
Drain	A 9.5	BOOL	Ansteuerung des Abflußventils
Drain_open_disp	A 9.6	BOOL	Anzeigelampe “Abflußventil offen”
Drain_closed_disp	A 9.7	BOOL	Anzeigelampe “Abflußventil geschlossen”

Symbolische Adressen der übrigen Programmelemente			
Symbolischer Name	**Adresse**	**Datentyp**	**Beschreibung**
EMER_STOP_off	E 1.6	BOOL	NOT-AUS-Schalter
Reset_maint	E 1.7	BOOL	Resettaster für die Wartungsan-zeigelampen aller Motoren
Motor_block	FB 1	FB 1	FB zum Steuern der Pumpen und des Rührwerkmotors
Valve_block	FC 1	FC 1	FC zum Steuern der Ventile
DB_feed_pump_A	DB 1	FB 1	Instanz-DB für die Steuerung der Speisepumpe A
DB_feed_pump_B	DB 2	FB 1	Instanz-DB für die Steuerung der Speisepumpe B
DB_agitator	DB 3	FB 1	Instanz-DB für die Steuerung des Rührwerkmotors

A.5.2.3 Erstellen des FB für den Motor

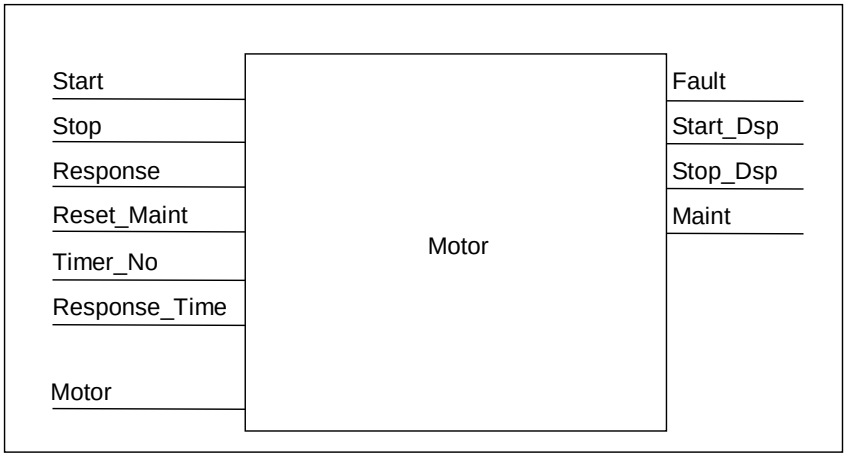
Anforderungen an den FB

Der FB für den Motor enthält die folgenden logischen Funktionen:

- Es gibt einen Start- und einen Stoppeingang.
- Verriegelungen ermöglichen den Betrieb der Geräte (Pumpen und Rührwerkmotor). Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z.B. “Enable_Motor”) und mit den Start- und Stoppeingängen verknüpft, wenn der FB für den Motor bearbeitet wird.
- Eine Rückmeldung von den Geräten muß innerhalb einer bestimmten Zeit erscheinen. Andernfalls wird angenommen, daß ein Fehler aufgetreten ist. Der FB stoppt daraufhin den Motor.
- Der Timer und die Zeitdauer für den Rückmelde- bzw. Fehlerzyklus müssen festgelegt werden.
- Ist der Starttaster gedrückt und die Freigabe erteilt, dann schaltet sich das Gerät ein und läuft solange, bis der Stopptaster gedrückt wird.
- Wird das Gerät eingeschaltet, wird ein Timer gestartet. Erscheint das Rückmeldesignal des Geräts nicht vor Ablauf des Timers, dann stoppt das Gerät.

Festlegen der Ein- und Ausgänge

Nachfolgendes Bild zeigt die Ein- und Ausgänge des allgemeinen FB für den Motor.



Definieren der Parameter für den FB

Wenn Sie einen wiederverwendbaren FB für den Motor (zum Steuern der beiden Pumpen und des Rührwerkmotors) erstellen möchten, müssen Sie allgemeine Parameternamen für die Eingänge und die Ausgänge definieren.

Der FB für den Motor im Beispielprozeß hat die folgenden Anforderungen:

- Er benötigt Signale vom Bedienpult zum Stoppen bzw. Starten des Motors und der Pumpen.
- Er benötigt ein Rückmeldesignal von Motor und Pumpen, daß der Motor läuft.
- Er muß die Zeit zwischen dem Senden des Signals, das den Motor einschaltet, und dem Empfangen des Rückmeldesignals ermitteln. Erscheint nach einer bestimmten Zeit kein Rückmeldesignal, muß der Motor ausgeschaltet werden.
- Er muß die entsprechenden Anzeigen auf dem Bedienpult ein- bzw. ausschalten.
- Er liefert ein Signal zur Ansteuerung des Motors.

Diese Anforderungen können als Eingänge und Ausgänge des FB festgelegt werden. Nachfolgende Tabelle zeigt die Parameter des FB für den Motor in unserem Beispielprozeß.

Parametername	Eingang	Ausgang	Durchgang
Start	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

Deklarieren der Variablen des FB für den Motor

Sie müssen die Eingangs-, Ausgangs- und Durchgangsparameter des FB für den Motor deklarieren.

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	IN	Start	BOOL	FALSE
0.1	IN	Stop	BOOL	FALSE
0.2	IN	Response	BOOL	FALSE
0.3	IN	Reset_Maint	BOOL	FALSE
2.0	IN	Time_No	TIMER	
4.0	IN	Response_Time	S5TIME	S5T#0MS
6.0	OUT	Fault	BOOL	FALSE
6.1	OUT	Start_Dsp	BOOL	FALSE
6.2	OUT	Stop_Dsp	BOOL	FALSE
6.3	OUT	Maint	BOOL	FALSE
8.0	IN_OUT	Motor	BOOL	FALSE
10.0	STAT	Time_bin	WORD	W#16#0
12.0	STAT	Time_BCD	WORD	W#16#0
14.0	STAT	Starts	INT	0
16.0	STAT	Start_Edge	BOOL	FALSE

Bei FBs werden die Eingangs-, Ausgangs-, Durchgangsparameter und die statischen Variablen in dem Instanz-DB gespeichert, der in der Aufrufanweisung angegeben wird. Die temporären Variablen werden im L-Stack gespeichert.

Programmieren des FB für den Motor

In STEP 7 muß jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. Im Beispielprogramm müssen Sie also den FB für den Motor vor dem OB 1 erstellen.

Der Anweisungsteil des FB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Start/Stop und Selbsthaltung

```
U(
O     #Start
O     #Motor
)
UN    #Stop
=     #Motor
```

Netzwerk 2 Anlaufüberwachung

```
U     #Motor
L     #Response_Time
SE    #Timer_No
UN    #Motor
R     #Timer_No
L     #Timer_No
T     #Timer_bin
LC    #Timer_No
T     #Timer_BCD
U     #Timer_No
UN    #Response
S     #Fault
R     #Motor
```

Netzwerk 3 Startlampe und Fehlerrücksetzung

```
U     #Response
=     #Start_Dsp
R     #Fault
```

Netzwerk 4 Stoplampe

```
UN    #Response
=     #Stop_Dsp
```

Netzwerk 5 Zählung der Starts

```
U     #Motor
FP    #Start_Edge
SPBN  lab1
L     #Starts
+     1
T     #Starts

lab1: NOP   0
```

Netzwerk 6 Wartungsanzeigelampe

```
L     #Starts
L     50
>=I
=     #Maint
```

Netzwerk 7 Rücksetzen des Zählers für die Anzahl der Starts

```
U     #Reset_Maint
U     #Maint
SPBN  END
L     0
T     #Starts

END: NOP   0
```

Erstellen der Instanz-Datenbausteine

Legen Sie drei Datenbausteine an und öffnen Sie diese nacheinander. Im Dialogfeld “Neuer Datenbaustein” wählen Sie das Optionsfeld “Datenbaustein mit zugeordnetem Funktionsbaustein”. Im Listenfeld “Zuordnung” wählen Sie “FB 1”. Damit sind die Datenbausteine als Instanz-Datenbausteine mit fester Zuordnung zum FB 1 festgelegt.

A.5.2.4 Erstellen der FC für die Ventile

Anforderungen an die FC

Die Funktion für die Einlaß- und Speiseventile sowie für das Abflußventil enthält die folgenden logischen Funktionen:

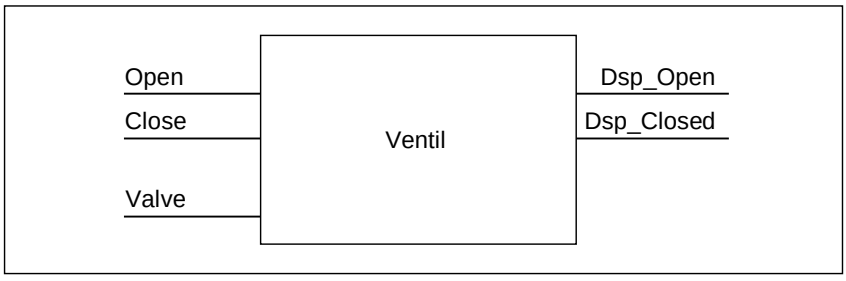
- Es gibt je einen Eingang zum Öffnen und zum Schließen der Ventile.
- Verriegelungen ermöglichen das Öffnen und Schließen der Ventile. Der Zustand der Verriegelungen wird in den temporären Lokaldaten (L-Stack) von OB 1 gespeichert (z.B. “Enable_Valve”) und mit den Eingängen zum Öffnen und Schließen verknüpft, wenn die FC für die Ventile bearbeitet wird.

Nachfolgende Tabelle zeigt die Parameter, die an die FC übergeben werden müssen.

Parameter für die Ventile	Eingang	Ausgang	Durchgang
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

Festlegen der Ein- und Ausgänge

Nachfolgende Bild zeigt die Ein- und Ausgänge der allgemeinen FC für die Ventile. Die Geräte, die den FB für den Motor aufrufen, übergeben Eingangsparameter. Die FC für die Ventile gibt Ausgangsparameter zurück.



Deklarieren der Variablen der FC für die Ventile

Wie bei dem FB für den Motor müssen Sie auch bei der FC für die Ventile die Eingangs-, Ausgangs- und Durchgangsparameter deklarieren (siehe nachfolgende Variablendeklarationstabelle).

Adresse	Deklaration	Name	Typ	Anfangswert
0.0	IN	Open	BOOL	FALSE
0.1	IN	Close	BOOL	FALSE
2.0	OUT	Dsp_Open	BOOL	FALSE
2.1	OUT	Dsp_Closed	BOOL	FALSE
4.0	IN_OUT	Valve	BOOL	FALSE

Bei FCs werden die temporären Variablen im L-Stack gespeichert. Die Eingangs-, Ausgangs- und Durchgangsvariablen werden als Pointer auf den Codebaustein gespeichert, der die FC aufgerufen hat. Zusätzlicher Speicherplatz im L-Stack (nach den temporären Variablen) wird für diese Variablen verwendet.

Programmieren der FC für die Ventile

Auch die Funktion FC 1 für die Ventile müssen Sie vor dem OB 1 erstellen, da aufgerufene Bausteine vor aufrufenden Bausteinen erstellt werden müssen.

Der Anweisungsteil des FC 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Öffnen/Schließen und Selbsthaltung

```
U(
O     #Open
O     #Valve
)
UN    #Close
=     #Valve
```

Netzwerk 2 Anzeige, ob Ventil geöffnet

```
U     #Valve
=     #Dsp_Open
```

Netzwerk 3 Anzeige, ob Ventil geschlossen

```
UN    #Valve
=     #Dsp_Closed
```

A.5.2.5 Erstellen des OB 1

OB 1 bestimmt die Struktur des Beispielprogramms. Außerdem enthält OB 1 die Parameter, die an die verschiedenen Bausteine übergeben werden, z. B.:

- Die AWL-Netzwerke für die Speisepumpen und den Rührwerkmotor liefern dem FB für den Motor die Eingangsparameter zum Starten (“Start”), Stoppen (“Stop”), für die Rückmeldung (“Response”) und für das Rücksetzen der Wartungsanzeige (”Reset_Maint”). Der FB für den Motor wird in jedem Zyklus des AS bearbeitet.
- Wird der FB für den Motor bearbeitet, legen die Eingänge “Timer_No” und “Response_Time” fest, welcher Timer verwendet wird und innerhalb welcher Zeitdauer eine Rückmeldung erfolgen muß.
- Die FC für die Ventile und der FB für die Motoren werden in jedem Programmzyklus des Automatisierungssystems bearbeitet, da sie im OB1 aufgerufen werden.

Das Programm verwendet den FB für den Motor mit verschiedenen Instanz-DBs, um die Aufgaben für die Steuerung der Speisepumpen und des Rührwerkmotors zu erfüllen.

Deklarieren von Variablen für OB 1

Nachfolgend wird die Variablendeklarationstabelle für den OB 1 abgebildet. Die ersten 20 Bytes enthalten die Startinformationen des OB 1 und dürfen nicht geändert werden.

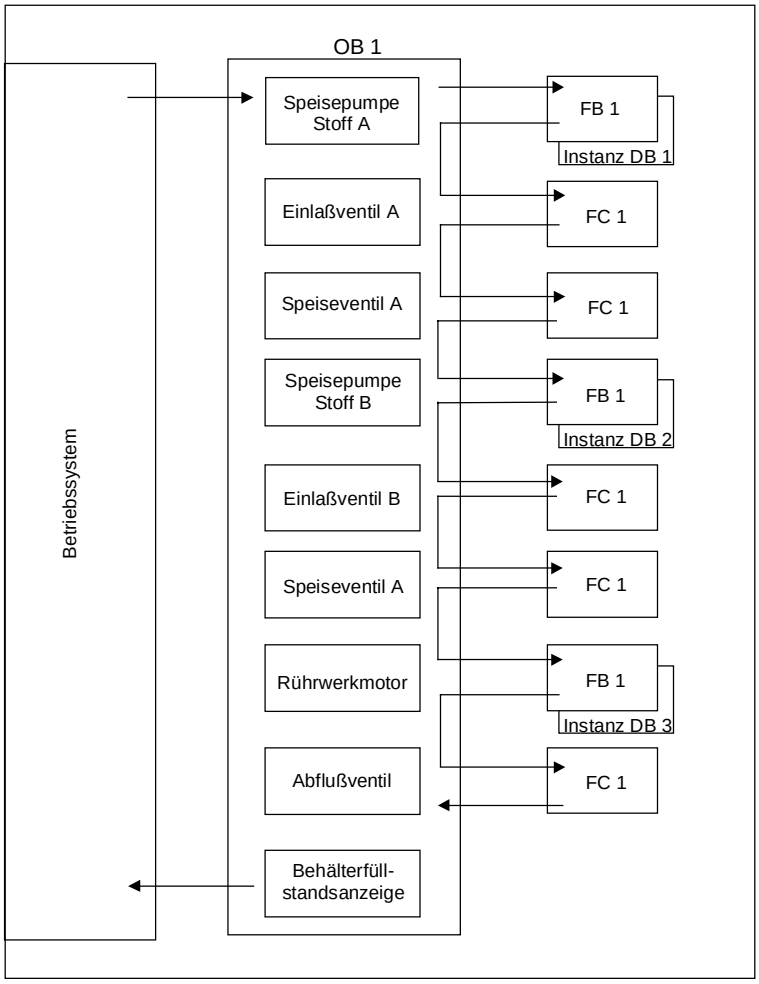
Adresse	Deklaration	Name	Typ
0.0	TEMP	OB1_EV_CLASS	BYTE
1.0	TEMP	OB1_SCAN1	BYTE
2.0	TEMP	OB1_PRIORITY	BYTE
3.0	TEMP	OB1_OB_NUMBR	BYTE
4.0	TEMP	OB1_RESERVED_1	BYTE
5.0	TEMP	OB1_RESERVED_2	BYTE
6.0	TEMP	OB1_PREV_CYCLE	INT
8.0	TEMP	OB1_MIN_CYCLE	INT
10.0	TEMP	OB1_MAX_CYCLE	INT
12.0	TEMP	OB1_DATE_TIME	DATE_AND_TIME
20.0	TEMP	Enable_Motor	BOOL
20.1	TEMP	Enable_Valve	BOOL
20.2	TEMP	Start_Fulfilled	BOOL
20.3	TEMP	Stop_Fulfilled	BOOL
20.4	TEMP	Inlet_Valve_A_Open	BOOL
20.5	TEMP	Inlet_Valve_A_Closed	BOOL
20.6	TEMP	Feed_Valve_A_Open	BOOL

Adresse	Deklaration	Name	Typ
20.7	TEMP	Feed_Valve_A_Closed	BOOL
21.0	TEMP	Inlet_Valve_B_Open	BOOL
21.1	TEMP	Inlet_Valve_B_Closed	BOOL
21.2	TEMP	Feed_Valve_B_Open	BOOL
21.3	TEMP	Feed_Valve_B_Closed	BOOL
21.4	TEMP	Open_Drain	BOOL
21.5	TEMP	Close_Drain	BOOL
21.6	TEMP	Close_Valve_Fulfilled	BOOL

Erstellen des Programms für OB 1

In STEP 7 muß jeder Baustein, der von einem anderen Baustein aufgerufen wird, vor dem Baustein erstellt werden, der den Aufruf enthält. In dem Beispielprogramm müssen Sie sowohl den FB für den Motor als auch die FC für die Ventile vor dem Programm in OB 1 erstellen.

Die Bausteine FB 1 und FC 1 werden im OB 1 mehrfach aufgerufen, der FB 1 mit verschiedenen Instanz-DBs:



Der Anweisungsteil des OB 1 sieht in der Programmiersprache AWL folgendermaßen aus:

Netzwerk 1 Verriegelungen für Speisepumpe A

```
U     ”EMER_STOP_off”
U     ”Tank_below_max”
UN    ”Drain”
=     #Enable_Motor
```

Netzwerk 2 Aufruf FB Motor für Stoff A

```
U               ”Feed_pump_A_start”
U               #Enable_Motor
=               #Start_Fulfilled
U(
O               ”Feed_pump_A_stop”
ON    #Enable_Motor
)
=               #Stop_Fulfilled
CALL  ”Motor_block”, ”DB_feed_pump_A”
      Start     :=#Start_Fulfilled
      Stop      :=#Stop_Fullfilled
      Response  :=”Flow_A”
      Reset_Maint   :=”Reset_maint”
      Timer_No  :=T12
      Reponse_Time:=S5T#7S
      Fault     :=”Feed_pump_A_fault”
      Start_Dsp :=”Feed_pump_A_on”
      Stop_Dsp  :=”Feed_pump_A_off”
      Maint     :=”Feed_pump_A_maint”
      Motor     :=”Feed_pump_A”
```

Netzwerk 3 Verzögerung der Ventilfreigabe Stoff A

```
U     ”Feed_pump_A”
L     S5T#1S
SE    T     13
UN    ”Feed_pump_A”
R     T     13
U     T     13
=     #Enable_Valve
```

Netzwerk 4 Einlaßventilsteuerung für Stoff A

```
UN    ”Flow_A”
UN    ”Feed_pump_A”
=               #Close_Valve_Fulfilled
CALL  ”Valve_block”
      Open      :=#Enable_Valve
      Close     :=#Close_Valve_Fulfilled
      Dsp_Open  :=#Inlet_Valve_A_Open
      Dsp_Closed:=#Inlet_Valve_A_Closed
      Valve     :=”Inlet_Valve_A”
```

Netzwerk 5 Speiseventilsteuerung für Stoff A

```
UN    ”Flow_A”
UN    ”Feed_pump_A”
=           #Close_Valve_Fulfilled
CALL  ”Valve_block”
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open      :=#Feed_Valve_A_Open
      Dsp_Closed:=#Feed_Valve_A_Closed
      Valve  :=”Feed_Valve_A”
```

Netzwerk 6 Verriegelungen für Speisepumpe B

```
U     ”EMER_STOP_off”
U     ”Tank_below_max”
UN    ”Drain”
=     ”Enable_Motor
```

Netzwerk 7 Aufruf FB Motor für Stoff B

```
U           ”Feed_pump_B_start”
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           ”Feed_pump_B_stop”
ON    #Enable_Motor
)
=           #Stop_Fulfilled
CALL  ”Motor_block”, ”DB_feed_pump_B”
      Start    :=#Start_Fulfilled
      Stop     :=#Stop_Fullfilled
      Response       :=”Flow_B”
      Reset_Maint    :=”Reset_maint”
      Timer_No       :=T14
      Reponse_Time:=S5T#7S
      Fault    :=”Feed_pump_B_fault”
      Start_Dsp      :=”Feed_pump_B_on”
      Stop_Dsp       :=”Feed_pump_B_off”
      Maint    :=”Feed_pump_B_maint”
      Motor    :=”Feed_pump_B”
```

Netzwerk 8 Verzögerung der Ventilfreigabe Stoff B

```
U     ”Feed_pump_B”
L     S5T#1S
SE    T     15
UN    ”Feed_pump_B”
R     T     15
U     T     15
=     #Enable_Valve
```

Netzwerk 9 Einlaßventilsteuerung für Stoff B

```
UN    ”Flow_B”
UN    ”Feed_pump_B”
=           #Close_Valve_Fulfilled
CALL  ”Valve_block”
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open      :=#Inlet_Valve_B_Open
      Dsp_Closed:=#Inlet_Valve_B_Closed
      Valve  :=”Inlet_Valve_B”
```

Netzwerk 10 Speiseventilsteuerung für Stoff B

```
UN    ”Flow_B”
UN    ”Feed_pump_B”
=           #Close_Valve_Fulfilled
CALL  ”Valve_block”
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open      :=#Feed_Valve_B_Open
      Dsp_Closed:=#Feed_Valve_B_Closed
      Valve  :=”Feed_Valve_B”
```

Netzwerk 11 Verriegelungen für Rührwerk

```
U     ”EMER_STOP_off”
U     ”Tank_above_min”
UN    ”Drain”
=     #Enable_Motor
```

Netzwerk 12 Aufruf FB Motor für Rührwerk

```
U           ”Agitator_start”
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           ”Agitator_stop”
ON    #Enable_Motor
)
=           #Stop_Fulfilled
CALL  ”Motor_block”, ”DB_Agitator”
      Start  :=#Start_Fulfilled
      Stop   :=#Stop_Fullfilled
      Response      :=”Agitator_running”
      Reset_Maint   :=”Reset_maint”
      Timer_No      :=T16
      Reponse_Time:=S5T#10S
      Fault  :=”Agitator_fault”
      Start_Dsp     :=”Agitator_on”
      Stop_Dsp      :=”Agitator_off”
      Maint  :=”Agitator_maint”
      Motor  :=”Agitator”
```

Netzwerk 13 Verriegelungen für Abflußventil

```
U     ”EMER_STOP_off”
U     ”Tank_not_empty”
UN    ”Agitator”
=     ”Enable_Valve
```

Netzwerk 14 Abflußventilsteuerung

```
U             ”Drain_open”
U             #Enable_Valve
=             #Open_Drain
U(
O             ”Drain_closed”
ON    #Enable_Valve
)
=             #Close_Drain
CALL  ”Valve_block”
      Open    :=#Open_Drain
      Close   :=#Close_Drain
      Dsp_Open      :=”Drain_open_disp”
      Dsp_Closed    :=”Drain_closed_disp”
      Valve   :=”Drain”
```

Netzwerk 15 Behälterfüllstandsanzeige

```
UN    ”Tank_below_max”
=     ”Tank_max_disp”
UN    ”Tank_above_min”
=     ”Tank_min_disp”
UN    ”Tank_not_empty”
=     ”Tank_empty_disp”
```

A.5.3 Beispiel zur Hantierung mit Uhrzeitalarmen

A.5.3.1 Struktur des Anwenderprogramms Uhrzeitalarme

Aufgabenstellung

Ausgang A 4.0 soll in der Zeit von Montag, 5.00 Uhr bis Freitag, 20.00 Uhr gesetzt sein. In der Zeit von Freitag, 20.00 Uhr bis Montag, 5.00 Uhr soll der Ausgang A 4.0 zurückgesetzt sein.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aufruf der Funktion FC 12
FC 12	In Abhängigkeit vom Zustand des Ausgangs A 4.0, des Uhrzeitalarm-Status und der Eingänge E 0.0 und E 0.1 • Startzeitpunkt vorgeben • Uhrzeitalarm setzen • Uhrzeitalarm aktivieren • CAN_TINT
OB 10	Abhängig vom aktuellen Wochentag • Startzeitpunkt vorgeben • Ausgang A 4.0 setzen bzw. rücksetzen • nächsten Uhrzeitalarm setzen • nächsten Uhrzeitalarm aktivieren
OB 80	Setzen des Ausgangs A 4.1 Startereignisinformation des OB 80 speichern im Merkerbereich

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten globalen Operanden. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von ”Uhrzeitalarm setzen” und Uhrzeitalarm aktivieren”
E 0.1	Eingang für die Stornierung eines Uhrzeitalarms
A 4.0	Ausgang, der vom Uhrzeitalarm-OB (OB 10) gesetzt/rückgesetzt wird
A 4.1	Ausgang, der bei einem Zeitfehler (OB 80) gesetzt wird
MW 16	STATUS des Uhrzeitalarms (SFC 31 ”QRY_TINT”)
MB 100 bis MB 107	Speicher für Startereignisinformation des OB 10 (nur Uhrzeit)
MB 110 bis MB 129	Speicher für Startereignisinformation des OB 80 (Zeitfehler)
MW 200	RET_VAL des SFC 28 ”SET_TINT”
MB 202	Binärergebnis- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 204	RET_VAL des SFC 30 ”ACT_TINT”
MW 208	RET_VAL des SFC 31 ”QRY_TINT”

Verwendete SFCs und FCs

Im Programmbeispiel werden die folgenden Systemfunktionen verwendet:

- SFC 28 ”SET_TINT” : Uhrzeitalarm stellen
- SFC 29 ”CAN_TINT” : Uhrzeitalarm stornieren
- SFC 30 ”ACT_TINT” : Uhrzeitalarm aktivieren
- SFC 31 ”QRY_TINT” : Uhrzeitalarm abfragen
- FC 3 ”D_TOD_DT” : DATE und TIME_OF_DAY zu DT zusammenfassen

A.5.3.2 FC 12

Deklarationsteil

Im Deklarationsteil des FC 12 werden folgende bausteintemporäre Variablen deklariert:

Variablenname	Datentyp	Deklaration	Kommentar
IN_UHRZEIT	TIME_OF_DAY	TEMP	Startzeit Vorgabe
IN_DATUM	DATE	TEMP	Startdatum Vorgabe
OUT_UHRZEIT_DATUM	DATE_AND_TIME	TEMP	Startdatum/-zeit gewandelt
OK_MERKER	BOOL	TEMP	Freigabe für Uhrzeitalarm-Stellen

Anweisungsteil AWL

Im Anweisungsteil des FC 12 geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (FC 12)                                  Erläuterung
Network 1:
       CALL SFC 31                           SFC QRY_TINT
        OB_NR  := 10                         STATUS Uhrzeitalarme abfragen.
        RET_VAL:= MW 208
        STATUS := MW 16

Network 2:
       UN     A      4.0                    Startzeitpunkt vorgeben in Abhaengigkeit
       SPB    mont                          von A 4.0 (in Variable
       L      D#1995-1-27                   #IN_DATUM und #IN_UHRZEIT)
       T      #IN_DATUM                     Startdatum ist ein Freitag.
       L      TOD#20:0:0.0
       T      #IN_UHRZEIT
       SPA    wndl
mont:  L      D#1995-1-23
       T      #IN_DATUM                     Startdatum ist ein Montag.
       L      TOD#5:0:0.0
       T      #IN_UHRZEIT
wndl:  NOP    0
```

```
AWL (FC 12)                                  Erläuterung
Network 3:
       CALL FC 3                             Format wandeln von DATE und TIME_OF_DAY
        IN1     := #IN_DATUM                 nach DATE_AND_TIME (für Uhrzeitalarm
        IN2     := #IN_UHRZEIT               stellen)
        RET_VAL := #OUT_UHRZEIT_DATUM
Network 4:
       U       E 0.0                         Alle Voraussetzungen für Uhrzeitalarm
       UN      M 17.2                        stellen erfüllt? (Eingang  für Freigabe
       U       M 17.4                        gesetzt und Uhrzeitalarm nicht aktiv und
       =       #OK_MERKER                    Uhrzeitalarm–OB ist geladen)
Network 5:                                   Wenn  ja, dann Uhrzeitalarm stellen...
       U       #OK_MERKER
       SPBNB   m001
       CALL SFC 28
        OB_NR   := 10
        SDT     := #OUT_UHRZEIT_DATUM
        PERIOD  := W#16#1201
        RET_VAL := MW 200
m001   U       BIE
       =       M 202.3                       ...und Uhrzeitalarm aktivieren.
Network 6:
       U       #OK_MERKER
       SPBNB   m002
       CALL SFC 30
        OB_NR   := 10
        RET_VAL := MW 204
m002   U       BIE
       =       M 202.4                       Wenn Eingang für die Stornierung von
Network 7:                                   Uhrzeitalarmen gesetzt ist, dann
       U       E 0.1                         Uhrzeitalarm stornieren.
       SPBNB   m003
       CALL SFC 29
        OB_NR   := 10
        RET_VAL := MW 210
m003   U       BIE
       =       M 202.5
```

A.5.3.3 OB 10

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 10 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)
- Weitere bausteintemporäre Variablen WTAG, IN_DATUM, IN_UHRZEIT und OUT_UHRZEIT_DATUM

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Gesamte Startereignisinformation des OB 10 als Struktur deklariert
E_ID	WORD	TEMP	Ereignis-ID
PR_KLASSE	BYTE	TEMP	Prioritätsklasse
OB_NR	BYTE	TEMP	OB-Nummer
RESERVED_1	BYTE	TEMP	Reserviert
RESERVED_2	BYTE	TEMP	Reserviert
PERIODE	WORD	TEMP	Periodizität des Uhrzeitalarms
RESERVED_3	DWORD	TEMP	Reserviert
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEK_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	
WTAG	INT	TEMP	Wochentag
IN_DATUM	DATE	TEMP	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
IN_UHRZEIT	TIME_OF_DAY	TEMP	Eingangsvariable für FC 3 (Wandlung des Zeitformates)
OUT_UHRZEIT_ DATUM	DATE_AND_TIME	TEMP	Ausgangsvariable für FC 3 und Eingangsvariable für SFC 28

Anweisungsteil AWL

Im Anweisungsteil des OB 10 geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (OB 10)                                    Erläuterung
Network 1:
       L      #STARTINFO.T_STMP.MSEK_WTAG      Wochentag selektieren
       L      W#16#F
       UW
       T      #WTAG                            und merken.
Network 2:
       L      #WTAG                            Wenn Wochentag nicht Montag ist, dann
       L      2                                Montag, 5.00 Uhr als nächsten
       <>I                                     Startzeitpunkt vorgeben und Ausgang  A
       SPB    mont                             4.0 rücksetzen.
Network 3:
       L      D#1995-1-27
       T      #IN_DATUM                        Sonst, d. h. wenn Wochentag = Montag
       L      TOD#20:0:0.0                     ist, dann Freitag, 20.00 Uhr als
       T      #IN_UHRZEIT                      nächsten Startzeitpunkt vorgeben und
       SET                                     Ausgang A 4.0 setzen.
       =      A 4.0
       SPA    wndl

mont:  L      D#1995-1-23
       T      #IN_DATUM
       L      TOD#5:0:0.0
       T      #IN_UHRZEIT
       CLR
       =      A 4.0

wndl:  NOP    0                                Startzeitpunkt vorgeben abgeschlossen.
Network 4:                                      Vorgegebenen Startzeitpunkt in Format
       CALL   FC 3                             DATE_AND_TIME wandeln (für SFC 28).
        IN1     := #IN_DATUM
        IN2     := #IN_UHRZEIT
        RET_VAL := #OUT_UHRZEIT_DATUM           Uhrzeitalarm stellen.
Network 5:
       CALL SFC 28
        OB_NR   := 10
        SDT     := #OUT_UHRZEIT_DATUM
        PERIOD  := W#16#1201
        RET_VAL := MW 200
       U      BIE
       =      M 202.1
Network 6:
       CALL SFC 30                              Uhrzeitalarm aktivieren.
        OB_NR   := 10
        RET_VAL := MW 204
       U      BIE
       =      M 202.2
Network 7:
       CALL SFC 20                              Blocktransfer: Uhrzeitangabe aus
        SRCBLK  := #STARTINFO.T_STMP            Startereignisinformation des OB 10 in
        RET_VAL := MW 206                       den Merkerbereich MB 100 bis MB 107
        DSTBLK  := P#M 100.0 BYTE  8            retten.
```

A.5.3.4 OB 1 und OB 80

Da die Startereignisinformation des OB 1 (OB für zyklisches Programm) in diesem Beispiel nicht ausgewertet wird, ist nur die Startereignisinformation des OB 80 dargestellt.

Anweisungsteil OB 1

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (OB 1)                                  Erläuterung
       CALL    FC 12                        Aufruf der Funktion FC 12
```

Deklarationsteil OB 80

Abweichend vom voreingestellten Deklarationsteil des OB 80 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Gesamte Startereignisinformation des OB 80 als Struktur deklariert
E_ID	WORD	TEMP	Ereignis-ID
PR_KLASSE	BYTE	TEMP	Prioritätsklasse
OB_NR	BYTE	TEMP	OB-Nummer
RESERVED_1	BYTE	TEMP	Reserviert
RESERVED_2	BYTE	TEMP	Reserviert
Z1_INFO	WORD	TEMP	Zusatzinformation über das Ereignis, das den Fehler verursacht hat.
Z2_INFO	DWORD	TEMP	Zusatzinformation über Ereignis-ID, Prioritätsklasse und OB-Nr. des Fehlerereignisses
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEK_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil OB 80

Im Anweisungsteil des OB 80, der vom Betriebssystem bei einem Zeitfehler aufgerufen wird, geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (OB 80)                              Erläuterung
Network 1:
       UN     A 4.1                      Ausgang A 4.1 setzen, wenn Zeitfehler
       S      A 4.1                      aufgetreten ist.

       CALL SFC 20                       Blocktransfer: Gesamte
        SRCBLK  := #STARTINFO            Startereignisinformation in den
        RET_VAL := MW 210                Merkerbereich MB 110 bis MB 129
        DSTBLK  := P#M 110.0 Byte 20     retten.
```

A.5.4 Beispiel zur Hantierung mit Verzögerungsalarmen

A.5.4.1 Struktur des Anwenderprogramms Verzögerungsalarme

Aufgabenstellung

Wenn Eingang E 0.0 gesetzt wird, soll Ausgang A 4.0 10 Sekunden später gesetzt werden. Jedes Setzen von Eingang E 0.0 soll die Verzögerungszeit neu starten.

Als anwendungsspezifisches Kennzeichen soll der Zeitpunkt (Sekunden- und Millisekunden) des Verzögerungsalarm-Starts in der Startereignisinformation des Verzögerungsalarm-OBs (OB 20) erscheinen.

Falls in diesen 10 Sekunden E 0.1 gesetzt wird, soll der Organisationsbaustein OB 20 nicht aufgerufen werden; d. h. der Ausgang A 4.0 soll nicht gesetzt werden.

Wenn Eingang E 0.2 gesetzt wird, soll Ausgang A 4.0 wieder zurückgesetzt werden.

Umsetzung im Anwenderprogramm

Nachfolgende Tabelle zeigt die Teilaufgaben der verwendeten Bausteine.

Baustein	Teilaufgabe
OB 1	Aktuelle Uhrzeit lesen und aufbereiten für Start des Verzögerungsalarms Abhängig vom Flankenwechsel am Eingang E 0.0 Verzögerungsalarm starten Abhängig vom Status des Verzögerungsalarms und vom Flankenwechsel am Eingang E 0.1 Verzögerungsalarm stornieren Abhängig vom Zustand des Eingangs E 0.2 Ausgang A 4.0 zurücksetzen
OB 20	Setzen des Ausgangs A 4.0 Aktuelle Uhrzeit lesen und aufbereiten Startereignisinformation in den Merkerbereich retten

Verwendete Operanden

Nachfolgende Tabelle zeigt die verwendeten Globaldaten. Die bausteintemporären Variablen sind im Deklarationsteil des jeweiligen Bausteins deklariert.

Operand	Bedeutung
E 0.0	Eingang für die Freigabe von ”Verzögerungsalarm starten”
E 0.1	Eingang für die Stornierung eines Verzögerungsalarms
E 0.2	Eingang für das Rücksetzen des Ausgangs A 4.0
A 4.0	Ausgang, der vom Verzögerungsalarm-OB (OB 20) gesetzt wird
MB 1	Genutzt für Flankenmerker und Binärergebnis- (Statusbit BIE)-Zwischenspeicher für SFCs
MW 4	STATUS des Verzögerungsalarms (SFC 34 ”QRY_TINT”)
MD 10	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 1
MW 100	RET_VAL des SFC 32 ”SRT_DINT”
MW 102	RET_VAL des SFC 34 ”QRY_DINT”
MW 104	RET_VAL des SFC 33 ”CAN_DINT”
MW 106	RET_VAL des SFC 20 ”BLKMOV”
MB 120 bis MB 139	Speicher für Startereignisinformation des OB 20
MD 140	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 20
MW 144	Sekunden und Millisekunden BCD-codiert aus der Startereignisinformation des OB 1; gewonnen aus Startereignisinformation des OB 20 (anwenderspezifisches Kennzeichen SIGN)

Verwendete SFCs

Im Anwenderprogramm ”Verzögerungsalarme” werden die folgenden Systemfunktionen verwendet:

- SFC 32 ”SRT_DINT” : Verzögerungsalarm starten
- SFC 33 ”CAN_DINT” : Verzögerungsalarm stornieren
- SFC 34 ”QRY_DINT” : Zustand eines Verzögerungsalarms abfragen

A.5.4.2 OB 20

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 20 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Start-Info zu OB 20
E_ID	WORD	TEMP	Ereignis-ID
AE_NR	BYTE	TEMP	Ablaufebene
OB_NR	BYTE	TEMP	OB-Nr
DK1	BYTE	TEMP	Datenkennung 1
DK2	BYTE	TEMP	Datenkennung 2
SIGN	WORD	TEMP	anwenderspezifisches Kennzeichen
DTIME	TIME	TEMP	Zeit, mit der Verzögerungsalarm gestartet wird
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben (Zeitstempel)
JAHR	BYTE	TEMP	
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEC_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil

Im Anweisungsteil des OB 20 geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (OB 20)                                    Erläuterung
Network 1:
       SET                                     Ausgang A 4.0 unbedingt setzen
       =       A 4.0
Network 2:
       L       AW 4                            Ausgangswort sofort aktualisieren
       T       PAW 4
Network 3:
       L       #STARTINFO.T_STMP.SEKUNDEN      Sekunden aus Startereignisinformation
       T       MW 140                          lesen
       L       #STARTINFO.T_STMP.MSEC_WTAG     Millisekunden und Wochentag aus
       T       MW 142                          Startereignsinformation lesen
       L       MD 140
       SRD     4                               Wochentag eliminieren und
       T       MD 140                          Millisekunden zurückschreiben (befinden
                                               sich nun BCD-codiert in MW 142).
Network 4:                                     Startzeitpunkt des Verzögerungsalarms
       L       #STARTINFO.SIGN                 (=Aufruf des SFC 32) aus
       T       MW 144                          Startereignisinformation lesen

Network 5:                                     Startereignisinformation in
       CALL SFC 20                             Merkerbereich kopieren (MB 120 bis MB
        SRCBLK  := STARTINFO                   139)
        RET_VAL := MW 106
        DSTBLK  := P#M 120.0 BYTE 20
```

A.5.4.3 OB 1

Deklarationsteil

Abweichend vom voreingestellten Deklarationsteil des OB 1 werden folgende bausteintemporäre Variablen deklariert:

- Struktur für gesamte Startereignisinformation (STARTINFO)
- Innerhalb der Struktur STARTINFO eine Struktur für die Uhrzeit (T_STMP)

Variablenname	Datentyp	Deklaration	Kommentar
STARTINFO	STRUCT	TEMP	Start-Info zu OB 1
E_ID	WORD	TEMP	Ereignis-ID
AE_NR	BYTE	TEMP	Ablaufebene
OB_NR	BYTE	TEMP	OB-Nr
DK 1	BYTE	TEMP	Datenkennung 1
DK 2	BYTE	TEMP	Datenkennung 2
AKT_ZYK	INT	TEMP	aktuelle Zykluszeit
MIN_ZYK	INT	TEMP	minimale Zykluszeit
MAX_ZYK	INT	TEMP	maximale Zykluszeit
T_STMP	STRUCT	TEMP	Struktur für Uhrzeitangaben (Zeitstempel)
JAHR	BYTE	TEMP	

Variablenname	Datentyp	Deklaration	Kommentar
MONAT	BYTE	TEMP	
TAG	BYTE	TEMP	
STUNDE	BYTE	TEMP	
MINUTEN	BYTE	TEMP	
SEKUNDEN	BYTE	TEMP	
MSEC_WTAG	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Anweisungsteil

Im Anweisungsteil des OB 1 geben Sie folgendes AWL-Anwenderprogramm ein:

```
AWL (OB 1)                                  Erläuterung
Network 1:
       L      #STARTINFO.T_STMP.SEKUNDEN    Sekunden aus Startereignisinformation
       T      MW 10                         lesen
       L      #STARTINFO.T_STMP.MSEC_WTAG   Millisekunden und Wochentag aus
       T      MW 12                         Startereignsinformation lesen
       L      MD 10                         Wochentag eliminieren und
       SRD    4                             Millisekunden zurückschreiben (befinden
       T      MD 10                         sich nun BCD-codiert in MW 12)
Network 2:                                  Positive Flanke an Eingang E 0.0?
       U      E 0.0
       FP     M 1.0
       =      M 1.1
Network 3:                                  Wenn ja, Verzögerungsalarm starten
       U      M 1.1                         (Startzeitpunkt des Verzögerungsalarms
       SPBNB  m001                          dem Parameter SIGN zugewiesen)
       CALL SFC 32
        OB_NR  := 20
        DTIME   := T#10S
        SIGN   := MW 12
        RET_VAL:= MW 100
m001:  NOP    0
                                            Status des Verzögerungsalarms abfragen
Network 4:                                  (SFC QRY_DINT)
       CALL SFC 34
        OB_NR  := 20
        RET_VAL:= MW 102
        STATUS := MW 4                      Positive Flanke am Eingang E 0.1?
Network 5:
       U      E 0.1
       FP     M 1.3
       =      M 1.4
                                            ... und Verzögerungsalarm ist aktiviert
Network 6:                                  (Bit 2 des Verzögerungsalarm-STATUS)?
       U      M 1.4                         Dann Verzögerungsalarm stornieren
       U      M 5.2
       SPBNB  m002
       CALL SFC 33
        OB_NR  := 20                        Ausgang A 4.0 mit Eingang E 0.2
        RET_VAL:= MW 104                    zurücksetzen
m002:  NOP    0
       U      E 0.2
       R      A 4.0
```

A.5.5 Beispiel für das Maskieren und Demaskieren von Synchronfehlerereignissen

Im folgenden Beispiel für ein Anwenderprogramm zeigen wir Ihnen das Maskieren und Demaskieren von Synchronfehlerereignissen. Mit dem SFC 36 ”MSK_FLT” werden in der Programmierfehlermaske folgende Fehler maskiert:

- Bereichslängenfehler beim Lesen
- Bereichslängenfehler beim Schreiben.

Mit einem 2. Aufruf des SFC 36 ”MSK_FLT” wird zusätzlich ein Zugriffsfehler maskiert:

- Peripheriezugriffsfehler beim Schreiben.

Mit dem SFC 38 ”READ_ERR” werden die maskierten Synchronfehlerereignisse abgefragt. Der ”Peripheriezugriffsfehler beim Schreiben” wird mit dem SFC 37 ”DMSK_FLT” wieder demaskiert.

Anweisung

Im folgenden sehen Sie den OB 1, in dem das Beispiel für das Anwenderprogramm in AWL programmiert wurde.

```
AWL (Network 1)                                  Erläuterung
       UN       M 255.0                          nicht remanenter Merker M 255.0 (nur
                                                 beim ersten Durchlauf=0)
       SPBNB    m001
       CALL     SFC 36                           SFC 36 MSK_FLT (Maskieren von
                                                 Synchronfehlerereignissen)
        PRGFLT_SET_MASK   :=DW#16#C               Bit2=Bit3=1 (BLFL und BLFS werden
                                                 maskiert)
        ACCFLT_SET_MASK   :=DW#16#0               alle Bits=0 (es wird kein Zugriffsfehler
                                                 maskiert)
        RET_VAL           :=MW 100                Returnwert
        PRGFLT_MASKED     :=MD 10                 Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 10
        ACCFLT_MASKED     :=MD 14                 Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 14
m001:  U         BIE
       S         M 255.0                          Setzen von M255.0, wenn Maskieren
                                                 erfolgreich
```

```
AWL (Network 2)                                  Erläuterung
       CALL     SFC 36                           SFC 36 MSK_FLT (Maskieren von
                                                 Synchronfehlerereignissen)
        PRGFLT_SET_MASK   :=DW#16#0               alle Bits=0 (es wird kein weiterer
                                                 Programmfehler maskiert)
        ACCFLT_SET_MASK   :=DW#16#8               Bit3=1 (schreibende Zugriffsfehler
                                                 werden maskiert)
        RET_VAL           :=MW 102                Returnwert
        PRGFLT_MASKED     :=MD 20                 Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 20
        ACCFLT_MASKED     :=MD 24                 Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 24
```

```
AWL (Network 3)                                  Erläuterung
       UN       M 27.3                           Baustein–Ende, wenn schreibender
       BEB                                       Zugriffsfehler (Bit3 in ACCFLT_MASKED)
                                                 nicht maskiert
```

```
AWL (Network 4)                                  Erläuterung
       L        B#16#0
       T        PAB 16                           Schreibender Zugriff (mit Wert 0) auf
                                                 PAB 16
```

```
AWL (Network 5)                                  Erläuterung
       CALL     SFC 38                           SFC 38 READ_ERR (Abfragen von
                                                 Synchronfehlerereignissen)
        PRGFLT_QUERY      :=DW#16#0              alle Bits=0 (es wird kein
                                                 Programmierfehler abgefragt)
        ACCFLT_QUERY      :=DW#16#8              Bit3=1 (schreibender Zugriffsfehler wird
                                                 abgefragt)
        RET_VAL           :=MW 104               Returnwert
        PRGFLT_CLR        :=MD 30                Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 30
        ACCFLT_CLR        :=MD 34                Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 34
       U        BIE                              kein Fehler aufgetreten und schreibenden
                                                 Zugriffsfehler festgestellt
       U        M 37.3
       NOT                                       VKE invertieren
       =        M 0.0                            M 0.0=1, wenn PAB 16 vorhanden
```

```
AWL (Network 6)                                  Erläuterung
       L        B#16#0
       T        PAB 17                           Schreibender Zugriff (mit Wert 0) auf
                                                 PAB 17
```

```
AWL (Network 7)                                  Erläuterung
       CALL     SFC 38                           SFC 38 READ_ERR (Abfragen von
                                                 Synchronfehlerereignissen)
        PRGFLT_QUERY      :=DW#16#0              alle Bits=0 (es wird kein
                                                 Programmierfehler abgefragt)
        ACCFLT_QUERY      :=DW#16#8              Bit3=1 (schreibender Zugriffsfehler wird
                                                 abgefragt)
        RET_VAL           :=MW 104               Returnwert
        PRGFLT_CLR        :=MD 30                Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 30
        ACCFLT_CLR        :=MD 34                Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 34
       U        BIE                              kein Fehler aufgetreten und schreibenden
                                                 Zugriffsfehler festgestellt
       U        M 37.3
       NOT                                       VKE invertieren
       =        M 0.1                            M 0.1=1, wenn PAB 17 vorhanden
```

```
AWL (Network 8)                                  Erläuterung
       L        B#16#0
       T        PAB 18                           Schreibender Zugriff (mit Wert 0) auf
                                                 PAB 18
```

```
AWL (Network 9)                                  Erläuterung
       CALL     SFC 38                           SFC 38 READ_ERR (Abfragen von
                                                 Synchronfehlerereignissen)
        PRGFLT_QUERY      :=DW#16#0              alle Bits=0 (es wird kein
                                                 Programmierfehler abgefragt)
        ACCFLT_QUERY      :=DW#16#8              Bit3=1 (schreibender Zugriffsfehler wird
                                                 abgefragt)
        RET_VAL           :=MW 104               Returnwert
        PRGFLT_CLR        :=MD 30                Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 30
        ACCFLT_CLR        :=MD 34                Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 34
       U        BIE                              kein Fehler aufgetreten und schreibenden
                                                 Zugriffsfehler festgestellt
       U        M 37.3
       NOT                                       VKE invertieren
       =        M 0.2                            M 0.2=1, wenn PAB 18 vorhanden
```

```
AWL (Network 10)                                 Erläuterung
       L        B#16#0
       T        PAB 19                           Schreibender Zugriff (mit Wert 0) auf
                                                 PAB 19
```

```
AWL (Network 11)                                 Erläuterung
       CALL     SFC 38                           SFC 38 READ_ERR (Abfragen von
                                                 Synchronfehlerereignissen)
        PRGFLT_QUERY      :=DW#16#0              alle Bits=0 (es wird kein
                                                 Programmierfehler abgefragt)
        ACCFLT_QUERY      :=DW#16#8              Bit3=1 (schreibender Zugriffsfehler wird
                                                 abgefragt)
        RET_VAL           :=MW 104               Returnwert
        PRGFLT_CLR        :=MD 30                Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 30
        ACCFLT_CLR        :=MD 34                Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 34
       U        BIE                              kein Fehler aufgetreten und schreibenden
                                                 Zugriffsfehler festgestellt
       U        M 37.3
       NOT                                       VKE invertieren
       =        M 0.3                            M 0.3=1, wenn PAB 19 vorhanden
```

```
AWL (Network 12)                                 Erläuterung
       CALL     SFC 37                           SFC 37 DMSK_FLT (Demaskieren von
                                                 Synchronfehlerereignissen)
        PRGFLT_RESET_MASK    :=DW#16#0           alle Bits=0 (es wird kein
                                                 Programmierfehler demaskiert)
        ACCFLT_RESET_MASK    :=DW#16#8           Bit3=1 (schreibender Zugriffsfehler wird
                                                 demaskiert)
        RET_VAL              :=MW 102            Returnwert
        PRGFLT_MASKED        :=MD 20             Ausgabe der aktuellen
                                                 Programmierfehlermaske in MD 20
        ACCFLT_MASKED        :=MD 24             Ausgabe der aktuellen
                                                 Zugriffsfehlermaske in MD 24
```

```
AWL (Network 13)                                 Erläuterung
       U        M 27.3                           Baustein–Ende, wenn schreibender
       BEB                                       Zugriffsfehler (Bit3 in ACCFLT_MASKED)
                                                 nicht demaskiert
```

```
AWL (Network 14)                                Erläuterung
        U       M 0.0
        SPBNB   m002
        L       EB 0                            EB 0 nach PAB 16 transferieren, wenn
        T       PAB 16                          vorhanden
m002:   NOP     0
```

```
AWL (Network 15)                                Erläuterung
        U       M 0.1
        SPBNB   m003
        L       EB 1                            EB 1 nach PAB 17 transferieren, wenn
        T       PAB 17                          vorhanden
m003:   NOP     0
```

```
AWL (Network 16)                                Erläuterung
        U       M 0.2
        SPBNB   m004
        L       EB 2                            EB 2 nach PAB 18 transferieren, wenn
        T       PAB 18                          vorhanden
m004:   NOP     0
```

```
AWL (Network 17)                                Erläuterung
        U       M 0.3
        SPBNB   m005
        L       EB 3                            EB 3 nach PAB 19 transferieren, wenn
        T       PAB 19                          vorhanden
m005:   NOP     0
```

A.5.5.1 Beispiel zum Sperren und Freigeben von Alarm- und Asynchronereignissen (SFC 39 und 40)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit dem SFC 39 ”DIS_IRT” OB 35-Aufrufe (Uhrzeitalarm) gesperrt und mit dem SFC 40 ”EN_IRT” werden die OB 35-Aufrufe wieder freigegeben.

Im OB 1 werden der SFC 39 und der SFC 40 aufgerufen:

```
AWL (OB 1)                                  Erläuterung
       U        M 0.0                       Programmteil, der problemlos
       S        M 90.1                      unterbrochen werden kann:
       U        M 0.1
       S        M 90.0
       :
       :                                    Programmteil, der nicht durch Interrupts
                                            unterbrochen werden darf:
       CALL     SFC 39                      Alarme sperren und verwerfen
        MODE    :=B#16#2                    Mode 2: einzelne Alarm-OBs sperren
        OB_NR   :=35                        OB35 sperren
        RET_VAL :=MW 100
       :
       :
       L        PEW 100
       T        MW 200
       L        MW 90
       T        MW 92
       :
       :
       CALL     SFC 40                      Alarme freigeben
        MODE    :=B#16#2                    Mode 2: einzelne Alarm-OBs freigeben
        OB_NR   :=35                        OB35 freigeben
        RET_VAL :=MW 102
                                            Programmteil, der problemlos
                                            unterbrochen werden kann:
       U        M 10.0
       S        M 190.1
       U        M 10.1
       S        M 190.0
       :
       :
```

A.5.6 Beispiel zur verzögerten Bearbeitung von Alarm- und Asynchronereignissen (SFC 41 und 42)

In diesem Beispiel für ein Anwenderprogramm wird ein Programmteil angenommen, der nicht durch Alarme unterbrochen werden darf. Für diesen Programmteil werden mit dem SFC 41 ”DIS_AIRT” Alarme verzögert und später mit dem SFC 42 ”EN_AIRT” wieder freigegeben.

In dem OB 1 werden der SFC 41 und der SFC 42 aufgerufen:

```
AWL (OB 1)                                  Erläuterung
       U        M 0.0                       Programmteil, der problemlos
       S        M 90.1                      unterbrochen werden kann:
       U        M 0.1
       S        M 90.0
       :
       :                                    Programmteil, der nicht durch Interrupts
                                            unterbrochen werden darf:
       CALL     SFC 41                      Alarme sperren und verzögern
        RET_VAL :=MW 100
       L        PEW 100
       T        MW 200
       L        MW 90
       T        MW 92
       :
       :
       :
       CALL     SFC 42                      Alarm freigeben
        RET_VAL :=MW 102
       L        MW 100                      Im Returnwert steht die Anzahl der
                                            eingelegten Alarmsperren
       DEC      1
       L        MW 102                      im Returnwert steht die Anzahl der
                                            eingelegten Alarmsperren
       <>I                                  die Anzahl muß nach der Alarmfreigabe
                                            den gleichen Wert haben
       SPB      fehl                        wie vor der Alarmsperre (hier ”0”)


       U        M 10.0                      Programmteil, der problemlos
       S        M 190.1                     unterbrochen werden kann:
       U        M 10.1
       S        M 190.0
       :
       :
       BEA
fehl:  L        MW 102                      Anzahl der eingelegten Alarmsperren wird
       T        AW 12                       angezeigt
```

A.6 Zugriff auf Prozeß- und Peripheriedatenbereiche

A.6.1 Zugriff auf den Prozeßdatenbereich

Die CPU kann entweder indirekt über das Prozeßabbild oder über den Rückwand-/P-Bus direkt auf Ein- und Ausgänge zentraler und dezentraler Digitalein-/ausgabebaugruppen zugreifen.

Auf Ein- und Ausgänge zentraler und dezentraler Analogein-/ausgabebaugruppen greift die CPU direkt über den Rückwand-/P-Bus zu.

Baugruppenadressierung

Die Zuordnung zwischen Adressen, die im Anwenderprogramm verwendet werden, und den Baugruppen erfolgt durch die Konfigurierung der Baugruppen mit STEP 7

- bei zentraler Peripherie: Anordnung des Baugruppenträgers und Zuordnung der Baugruppen auf Steckplätzen in der Konfigurationstabelle
- bei dezentraler Peripherie (PROFIBUS-DP): Anordnung der DP-Slaves in der Konfigurationstabelle ”Mastersystem” mit Vergabe der PROFIBUS-Adresse und Zuordnung der Baugruppen auf Steckplätze.

Das Konfigurieren der Baugruppen löst die Adreßeinstellung der einzelnen Baugruppen über Schalter ab. Die CPU erhält vom PG als Ergebnis der Konfigurierung Daten, anhand derer sie die zugeordneten Baugruppen erkennt.

Peripherieadressierung

Für Ein- und Ausgänge existiert jeweils ein eigener Adreßbereich. Deshalb muß die Adresse eines Peripheriebereichs außer der Byte- oder Wortangabe zusätzlich die Kennung E für die Eingänge und A für die Ausgänge beinhalten.

Nachfolgende Tabelle zeigt die zur Verfügung stehenden Peripherie-Adreßbereiche.

Operandenbereich	Zugriff über Einheiten der folgenden Größe:	S7-Notation
Peripheriebereich: Eingänge	Peripherieeingangsbyte Peripherieeingangswort Peripherieeingangs-Doppelwort	PEB PEW PED
Peripheriebereich: Ausgänge	Peripherieausgangsbyte Peripherieausgangswort Peripherieausgangs-Doppelwort	PAB PAW PAD

Welche Adreßbereiche bei den einzelnen Baugruppen möglich sind, entnehmen Sie den folgenden Handbüchern.

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

Baugruppen-Anfangsadresse

Die Baugruppen-Anfangsadresse ist die niedrigste Byte-Adresse einer Baugruppe. Sie stellt die Anfangsadresse des Nutzdatenbereichs der Baugruppe dar und wird in vielen Fällen stellvertretend für die ganze Baugruppe verwendet.

Sie wird z. B. bei Prozeßalarmen, Diagnosealarmen, Ziehen/Stecken-Alarmen, und Stromversorgungsfehlern in die Startinformation des zugehörigen Organisationsbausteins eingetragen und identifiziert damit die alarmgebende Baugruppe.

A.6.2 Zugriff auf den Peripheriedatenbereich

Der Peripheriedatenbereich läßt sich unterteilen in:

- Nutzdaten und
- Diagnose- und Parameterdaten.

Beide Bereiche haben einen Eingangsbereich (nur lesender Zugriff möglich) und einen Ausgangsbereich (nur schreibender Zugriff möglich).

Nutzdaten

Adressiert werden die Nutzdaten über die Byteadresse (bei digitalen Signalbaugruppen) oder die Wortadresse (bei analogen Signalbaugruppen) des Ein- bzw. Ausgangsbereichs. Auf Nutzdaten zugreifen können Sie über Lade- und Transferbefehle, Kommunikationsfunktionen (BuB-Zugriffe) bzw. über den Prozeßabbildtransfer. Nutzdaten können sein:

- digitale und analoge Ein-, Ausgangssignale von Signalbaugruppen
- Steuer- und Statusinformationen von Funktionsbaugruppen und
- Informationen für Punkt-zu-Punkt und Buskopplungen von Kommunikationsbaugruppen (nur S7-300).

Bei der Übertragung von Nutzdaten kann eine Datenkonsistenz von maximal 4 Byte erreicht werden (Ausnahme DP-Normslaves, siehe "Einstellen des Betriebsverhaltens"). Verwenden Sie die Anweisung ”Transferiere Doppelwort”, werden 4 Byte zusammenhängend und unverändert (konsistent) übertragen. Verwenden Sie vier einzelne Anweisungen ”Transferiere Eingangsbyte”, könnte an einer Befehlsgrenze ein Prozeßalarm-OB gestartet werden, der Daten an die gleiche Adresse überträgt und damit den Inhalt der ursprünglichen vier Byte verändert.

Diagnose- und Parameterdaten

Die Diagnose- und Parameterdaten einer Baugruppe können nicht einzeln adressiert werden, sondern nur zusammengefaßt zu ganzen Datensätzen. Diagnose- und Parameterdaten werden grundsätzlich konsistent übertragen.

Adressiert werden die Diagnose- und Parameterdaten über die Anfangsadresse der betreffenden Baugruppe und die Datensatznummer. Datensätze werden unterteilt in Eingangs- und Ausgangsdatensätze. Eingangsdatensätze können nur gelesen, Ausgangsdatensätze können nur beschrieben werden. Auf Datensätze zugreifen können Sie mit Hilfe von Systemfunktionen oder Kommunikationsfunktionen (Bedienen und Beobachten, BuB). Nachfolgende Tabelle zeigt die Zuordnung der Datensätze zu Diagnose- und Parameterdaten.

Daten	Beschreibung
Diagnosedaten	Bei diagnosefähigen Baugruppen erhalten Sie beim Lesen der Datensätze 0 und 1 die Diagnosedaten dieser Baugruppe.
Parameterdaten	Bei parametrierbaren Baugruppen übertragen Sie beim Schreiben der Datensätze 0 und 1 die Parameter dieser Baugruppe.

Zugreifen auf Datensätze

Die Informationen in den Datensätzen einer Baugruppe können Sie nutzen, um parametrierbare Baugruppen nachzuparametrieren und Diagnoseinformationen diagnosefähiger Baugruppen auszulesen.

Nachfolgende Tabelle zeigt, mit welchen Systemfunktionen Sie auf Datensätze zugreifen können.

SFC	Anwendung
Baugruppen parametrieren	
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den SDBs 100 bis 129 zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe
Diagnoseinformation auslesen	
SFC 59 RD_REC	Auslesen der Diagnosedaten

Adressieren von S5-Baugruppen

Sie haben die Möglichkeit:

- Erweiterungsgeräte der SIMATIC S5 mit der Anschaltungsbaugruppe IM 463-2 an eine S7-400 zu koppeln und
- einige S5-Baugruppen in Adaptionskapseln in den zentralen Baugruppenträger der S7-400 zu stecken.

Wie Sie die S5-Baugruppen bei SIMATIC S7 adressieren, entnehmen Sie bitte dem Installationshandbuch "Automatisierungssysteme S7-400, M7-400, Aufbauen" bzw. der mitgelieferten Beschreibung der Adaptionskapsel.

A.7 Einstellen des Betriebsverhaltens

A.7.1 Einstellen des Betriebsverhaltens

Dieses Kapitel erläutert, wie Sie die Eigenschaften von S7-300- und S7-400-Automatisierungssystemen, die nicht fest vorgegeben sind, durch Einstellen von Systemparametern oder Einsatz von Systemfunktionen SFC beeinflussen können.

Detaillierte Informationen zu den Baugruppenparametern finden Sie in der STEP 7-Online-Hilfe sowie in folgenden Handbüchern:

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

Alles über SFCs finden Sie im Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Adressierung von DP-Normslaves

Sollen DP-Normslaves Daten größer 4 Byte übertragen oder empfangen, müssen Sie für die Übertragung dieser Daten spezielle SFCs verwenden.

SFC	Anwendung
Baugruppen parametrieren	
SFC 15 DPWR_DAT	Übertragen beliebiger Daten zur adressierten Signalbaugruppe
Diagnoseinformation auslesen	
SFC 13 DPNRM_DG	Auslesen der Diagnosedaten (asynchroner Lesevorgang)
SFC 14 DPRD_DAT	Auslesen konsistenter Daten (Länge 3 oder größer 4 Byte)

Beim Eintreffen eines DP-Diagnosetelegramms wird ein Diagnosealarm mit 4 Byte Diagnosedaten an die CPU gemeldet. Diese 4 Byte können mit der SFC 13 DPNRM_DG ausgelesen werden.

A.7.2 Ändern des Verhaltens und der Eigenschaften von Baugruppen

Defaulteinstellungen

- Alle parametrierbaren Baugruppen des S7-Automatisierungssystems sind bei Lieferung auf Defaultwerte eingestellt, die für Standardanwendungen geeignet sind. Mit diesen Defaultwerten können Sie die Baugruppen direkt, ohne weitere Einstellungen, einsetzen. Die Defaultwerte entnehmen Sie den Baugruppen-Beschreibungen folgender Handbücher:
- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

Welche Baugruppen können parametriert werden?

Sie können jedoch auch das Verhalten und die Eigenschaften der Baugruppen parametrieren und damit auf Ihre Erfordernisse und die Gegebenheiten Ihrer Anlage einstellen. Parametrierbare Baugruppen sind CPUs, FMs, CPs sowie einige Analogein-/-ausgabebaugruppen und Digitaleingabebaugruppen.

Es gibt parametrierbare Baugruppen mit und ohne Pufferung.

Baugruppen ohne Pufferung müssen nach jedem Spannungsausfall erneut mit den entsprechenden Daten versorgt werden. Die Parameter dieser Baugruppen werden im remanenten Speicherbereich der CPU gespeichert (indirekte Parametrierung durch die CPU).

Einstellen und Laden der Parameter

Baugruppenparameter stellen Sie mit STEP 7 ein. Beim Speichern der Parameter erzeugt STEP 7 das Objekt "Systemdatenbausteine", das mit dem Anwenderprogramm in die CPU geladen und von dort beim Anlauf in die zugehörigen Baugruppen übertragen wird.

Was kann parametriert werden?

Die Baugruppenparameter werden eingeteilt in Parameterblöcke. Welche Parameterblöcke auf welcher CPU verfügbar sind, entnehmen Sie bitte dem Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und dem Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten".

Beispiele für Parameterblöcke:

- Anlaufverhalten
- Zyklus
- MPI
- Diagnose
- Remanenz
- Taktmerker
- Alarmbehandlung
- On-Board-Peripherie (nur für S7-300)
- Schutzstufe
- Lokaldaten
- Echtzeituhr
- Asynchronfehler

Parametrieren mit SFCs

Außer der Parametrierung mit STEP 7 gibt es noch die Möglichkeit, mit Hilfe von Systemfunktionen Baugruppenparameter vom S7-Programm aus zu ändern. Nachfolgende Tabelle zeigt, welche SFCs welche Baugruppenparameter.

SFC	Anwendung
SFC 55 WR_PARM	Übertragen der änderbaren Parameter (Datensatz 1) zur adressierten Signalbaugruppe.
SFC 56 WR_DPARM	Übertragen der Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 57 PARM_MOD	Übertragen aller Parameter aus den zugehörigen SDBs zur adressierten Signalbaugruppe.
SFC 58 WR_REC	Übertragen eines beliebigen Datensatzes zur adressierten Signalbaugruppe

Die Systemfunktionen sind ausführlich beschrieben im Referenzhandbuch "Systemsoftware für S7-300/400, System- und Standardfunktionen".

Welche Baugruppenparameter dynamisch änderbar sind, entnehmen Sie bitte den folgenden Handbüchern:

- Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten"
- Referenzhandbuch "Automatisierungssysteme S7-300, M7-300, Baugruppendaten"
- Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten"

A.7.3 Nutzen der Uhrzeitfunktionen

Alle S7-300-/S7-400-CPUs sind mit einer Uhr (Echtzeituhr oder Software-Uhr) ausgestattet. Die Uhr kann im Automatisierungssystem sowohl als Uhrzeitmaster als auch als Slave mit externer Synchronisation fungieren. Sie ermöglicht die Verwendung von Uhrzeitalarmen und Betriebsstundenzählern.

Uhrzeitformat

Die Uhr zeigt immer Uhrzeit (Mindestauflösung 1 s) und Datum mit Wochentag an. Bei einigen CPUs ist auch die Anzeige von Millisekunden möglich (siehe Handbuch "Automatisierungssystem S7-300, Aufbauen, CPU-Daten" und Referenzhandbuch "Automatisierungssysteme S7-400, M7-400, Baugruppendaten").

Stellen und Lesen der Uhrzeit

Sie stellen die Uhrzeit und das Datum der CPU-Uhr mit dem Aufruf der SFC 0 SET_CLK aus dem Anwenderprogramm oder über Menübefehl vom PG aus und starten damit die Uhr. Mit der SFC 1 READ_CLK oder über Menübefehl vom PG aus lesen Sie das aktuelle Datum und die Uhrzeit der CPU aus.

Hinweis: Um Anzeigedifferenzen in HMI-Systemen in der Uhrzeit zu vermeiden, sollten Sie auf der CPU **Winterzeit** einstellen!

Parametrieren der Uhr

Ist in einem Netz mehr als eine Baugruppe mit Uhr vorhanden, müssen Sie durch Parametrieren mit STEP 7 einstellen, welche CPU bei der Synchronisation der Uhrzeit als Master und welche als Slave fungieren soll. Durch Parametrierung stellen Sie außerdem ein, ob die Synchronisierung über den K-Bus oder über die MPI-Schnittstelle erfolgen soll und in welchen Intervallen die automatische Synchronisierung erfolgen soll.

Synchronisieren der Uhrzeit

Um sicherzustellen, daß die Uhrzeit aller Baugruppen im Netz übereinstimmt, werden die Slave-Uhren vom Systemprogramm in regelmäßigen (parametrierbaren) Abständen synchronisiert. Mit der Systemfunktion SFC 48 SNC_RTCB können Sie Datum und Uhrzeit von der Master-Uhr an die Slave-Uhren übertragen.

Einsetzen eines Betriebsstundenzählers

Ein Betriebsstundenzähler zählt die Einschaltzeiten eines angeschlossenen Betriebsmittels oder die Betriebsdauer der CPU als Summe der Betriebsstunden.

Im Betriebszustand STOP wird der Betriebsstundenzähler angehalten. Sein Wert bleibt auch nach Urlöschen erhalten. Bei Neustart muß der Betriebsstundenzähler vom Anwenderprogramm wieder gestartet werden, bei Wiederanlauf läuft er automatisch weiter, wenn er vorher gestartet war.

Mit der SFC 2 SET_RTM können Sie den Betriebsstundenzähler auf einen Anfangswert setzen. Mit der SFC 3 CTRL_RTM können Sie den Betriebsstundenzähler starten oder stoppen.

Mit der SFC 4 READ_RTM können Sie die aktuelle Anzahl der Betriebsstunden und den Zustand des Zählers (”gestoppt” oder ”zählt”) auslesen.

Eine CPU kann bis zu 8 Betriebsstundenzähler haben. Die Numerierung beginnt bei 0.

A.7.4 Verwenden von Taktmerkern und Zeiten

Taktmerker

Ein Taktmerker ist ein Merker, der seinen Binärzustand periodisch im Puls-Pausen-Verhältnis 1:1 ändert. Welches Merkerbyte der CPU zum Taktmerkerbyte wird, bestimmen Sie bei der Parametrierung des Taktmerkers mit STEP 7.

Nutzen

Taktmerker können Sie im Anwenderprogramm verwenden, um z. B. Leuchtmelder mit Blinklicht anzusteuern oder periodisch wiederkehrende Vorgänge (etwa das Erfassen eines Istwertes) anzustoßen.

Mögliche Frequenzen

Jedem Bit des Taktmerkerbyte ist eine Frequenz zugeordnet. Nachfolgende Tabelle zeigt die Zuordnung:

Bit des Taktmerkerbytes	7	6	5	4	3	2	1	0
Periodendauer (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frequenz (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Hinweis

Taktmerker laufen asynchron zum CPU-Zyklus, d. h. in langen Zyklen kann sich der Zustand des Taktmerkers mehrfach ändern.

Zeiten

Zeiten sind ein Speicherbereich des Systemspeichers. Die Funktion einer Zeit wird durch das Anwenderprogramm festgelegt (z. B. Einschaltverzögerung). Die Anzahl der verfügbaren Zeiten ist CPU-abhängig).

Hinweis

- Wenn Sie in Ihrem Anwenderprogramm mehr Zeiten einsetzen, als die CPU zuläßt, wird ein synchroner Fehler gemeldet und OB 121 gestartet.
- Bei S7-300 (Ausnahme CPU 318) können Zeiten nur im OB 1 und im OB 100 gleichzeitig gestartet und aktualisiert werden, in allen anderen OBs können Zeiten nur gestartet werden.

Index

Abfragen des Uhrzeitalarms 4-25
Ablaufsteuerung 8-7
Absolute und symbolische Adressierung..... 7-1
ACT_TINT 4-25, 4-26
Adreßbelegung prüfen 2-12
Adressen von Kommunikationsteilnehmern
  einstellen 17-8
Adressierung
  absolut 7-1, 7-2
  bereichsintern A-52
  bereichsübergreifend A-52
  speicherindirekt A-52
  symbolisch 7-1
  von S5-Baugruppen A-113
Aktivieren
  Anzeige von Symbolen im Baustein 7-11
Aktualisieren
  Bausteinaufrufe 9-23
  Prozeßabbild 4-15
  Prozeßabbild A-18, A-19, A-20
Aktualparameter 4-16, 4-17
Aktuelle Baugruppen 1-10
Alarmarten 4-4
Alarmereignisse
  sperren und freigeben A-108
  verzögert bearbeiten A-109
Alarmgesteuerte Programmbearbeitung 4-3
Alarm-OBs
  abwählen 4-6, 4-7
  Einsatz 4-24
  parametrieren 4-3, 4-25
Allgemeine Hinweise
  zu Variablendeklarationstabellen 9-8
  zur Eingabe von Symbolen 7-10
Ältere Projekte verwenden A-70, A-71
Ändern
  Betriebszustands 16-4
  Datenwerte in der Datensicht von
    Datenbausteinen 10-8
  Uhrzeit für Uhrzeitalarm 4-25
Ändern der Fensteranordnung von
  Symboltabellen 5-28
Ändern der PROFIBUS-Adresse bei
  DP-Slaves 17-9
Ändern der Teilnehmeradresse bei
  S7-Stationen 17-9
Ändern des Verhaltens und der
  Eigenschaften von Baugruppen A-115
Ändern von BuB-Attributen mit CFC 15-4
Anfangsadresse A-111
ANLAUF A-8, A-9, A-10, A-11
  abbrechen A-5
  Betriebszustand der CPU A-1
  CPU-Tätigkeiten A-5
Anlauf-OB A-5, A-9, A-11
Anlauf-OBs 4-30
  Soll-Ist-Baugruppenüberwachung 4-30
  Startereignisse 4-31
Anlaufprogramm 4-31
Anlegen
  AWL-Quellen 11-13
  Referenzdaten 12-11
Anlegen eines Projekts 6-3, 6-4
Anordnen
  Boxen 9-20, 9-21
Anweisungen eingeben
  Vorgehensweise 9-12
Anweisungen aus dem Programmelemente-
  Katalog 9-4
Anweisungsliste 8-5
Anweisungsteil 9-1, 9-6
  Aufbau 9-11
  bearbeiten 9-11
  in KOP 9-5
  Suchfunktion für Fehler 9-14
Anwenderdefinierte Datentypen
  Beschreibung A-47
  erstellen A-47, A-48
  Struktur eingeben 10-6
Anwenderdefinierte Datentypen (UDT) 8-11
Anwenderdefinierte Diagnosemeldungen
  anlegen und bearbeiten 14-15
  anzeigen 14-20
Anwenderprogramm
  Aufgaben 4-1, 4-2
  Elemente 4-2
  im CPU-Speicher A-14
  laden A-14, A-15
Anwenderprogramme
  laden in Zielsystem 17-3
Anwenderspeicher
  komprimieren 17-20, 17-21
Anwendertexte
  siehe Bedienerrelevante Texte 14-16
Anwendertexte übersetzen und bearbeiten 14-16

Anwendungsfälle für Speichern / Archivieren 22-5
ANY A-50, A-56, A-57, A-58
Parameter
Beschreibung und Verwendung..... A-58
Anzeige
als Baumstruktur..... 12-4
als paarweise Aufrufbeziehung..... 12-6
Baustein-Längen..... 8-14
Datenstruktur von Datenbausteinen mit zugeordnetem FB (Instanz-DBs)..... 10-5
gelöschter Baustein..... 12-4
maximaler Lokaldatenbedarf in der Baumstruktur 12-4
sprachabhängiger Informationen zu KOP, FUP, AWL 12-9
Symbole im Baustein aktivieren..... 7-11
Anzeigeelemente beschreiben 3-9
Anzeigemöglichkeiten
für CPU-Meldungen und anwenderdefinierte Diagnosemeldungen 14-20
Anzeigen
Baugruppenzustand..... 21-2
Baustein-Längen..... 8-14
Betriebszustand..... 16-4
fehlende Symbole 12-10
im Programmstatus..... 19-3
Listen in zusätzlichen Arbeitsfenstern 12-10
nicht verwendete Operanden.....12-10
Operanden ohne Symbol.....12-10
Programmstruktur 12-11
Referenzdaten 12-10, 12-11, 12-12
Struktur von Datenbausteinen mit zugeordnetem UDT..... 10-7
Anzeigen von gespeicherten CPU-Meldungen 14-22
Anzeigesprache.....14-16
Arbeiten
mit Bibliotheken..... 8-17
Arbeitspeicher A-14
Arbeitsspeicher.....17-3, 17-4, A-13
Archiv
CPU-Meldungen 14-20, 14-21
Archivieren
Anwendungsfälle..... 22-5
Projekte und Bibliotheken..... 22-4
Voraussetzungen..... 22-6
Vorgehensweise 22-6
ARRAY..... A-39
Assistent zum unterstützten Anlegen eines Projekts 6-3
asynchrone Ereignisse..... 4-11
Asynchrone Fehler
OB81 21-24
Verwenden von OBs um auf Fehler zu reagieren 4-33
Asynchronereignisse
sperren und freigeben.....A-108
verzögert bearbeiten.....A-109
Attribute für Bausteine und Parameter.....8-16
Aufbau
Anweisungsteil 9-11
Fenster.....5-19
Online-Verbindung über das Fenster "Erreichbare Teilnehmer" 16-1
Online-Verbindung über das Online-Fenster des Projekts.....16-2
Online-Verbindungen.....16-1
Querverweisliste..... 12-2, 12-3
UDT 8-11
Variablendeklarationstabelle 9-6, 9-7
Auflisten von Ein- Aus- und Durchgängen 3-6
Aufruf Baugruppenzustand aus der Projektansicht (Online).....21-6
Aufrufen der Hilfefunktionen.....5-4
Aufrufhierarchie im Anwenderprogramm 4-9
Aufrufmöglichkeiten des Baugruppenzustands 21-9
Aufrufumgebung des Bausteins festlegen....19-9
Aufrufumgebung für Multiinstanzaufrufe festlegen 19-9
Aufteilung der Speicherbereiche A-13
Ausdrucken
zur Projektdokumentation 22-1
Ausgänge
auflisten 3-6
Belegungsplan.....12-6
Prozeßabbild..... A-18
Ausgangsdiagramm für die Motoren erstellen 3-6
Ausgangsdiagramm für die Ventile erstellen..3-7
Ausgangsparameter
RET_VAL auswerten 21-23
Auskunftsfunktionen 1-10, 21-12
Auskunftsfunktionen der Schnellansicht 21-5
Auskunftsfunktionen des Baugruppen-zustands 21-10
Auswahl des Meldeverfahrens 14-2
Auswahl von Objekten im Dialog.....5-26
Auswahldialog.....5-26
Auswählen
Programmerstellmethode.....8-1
Programmiersprache 8-2, 8-3
Auswerten
Ausgangsparameter RET_VAL .. 21-23, 21-24
Auswertung des Diagnosepuffers..... A-24
AuthorsW.....2-2

AuthorsW.exe 2-1
Automatisierungslösung konzipieren
Auflisten von Ein- Aus- und Durchgängen . 3-6
Beschreiben der einzelnen Funktionsbereiche 3-4
Beschreiben der notwendigen Anzeige- und Bedienelemente 3-9
Eingangs-/ Ausgangs-Diagramms für die Motoren erstellen 3-6
Erstellen eines Eingangs-/ Ausgangs-Diagramms für die Ventile..... 3-7
Konfigurationsplan erstellen 3-10
Prinzipielle Vorgehensweise 3-1
Sicherheitsanforderungen festlegen..... 3-7
Zerlegen des Prozesses in Aufgaben und Bereiche..... 3-2
Autorisierung 2-1
Anzahl 2-7
deinstallieren 2-1, 2-2, 2-5
Erstinstallation 2-3
hochrüsten.....2-3, 2-4
installieren 2-1, 2-2
Originaldiskette 2-1
Regeln..... 2-5
später durchführen..... 2-3
übertragen 2-3, 2-5
verlieren 2-1
wiederherstellen..... 2-4
Autorisierungsdiskette..... 2-1, 2-2, 2-3, 2-4, 2-5
Autorisierungsprogramm..... 2-3
AWL.....8-2, 8-3, 8-5
Anzeige sprachabhängiger Information... 12-9
Bausteine eingeben 9-12
Einstellungen 9-22
AWL-Anweisungen
Regeln für die Eingabe..... 9-22
Regeln zur Eingabe..... 11-2
AWL-Quelle
Formate für Bausteine.....11-10
AWL-Quellen
anlegen 11-13
aus Bausteinen generieren.....11-15
Bausteinvorlagen einfügen 11-14
Beispiel für FBs.....11-22
Beispiel für UDTs.....11-24
Beispiele für DBs 11-23
Beispiele für Variablendeklaration.....11-18
externe Quellen einfügen 11-14
Fehlersuche.....11-16
Grundlage zum Programmieren..... 11-1
Konsistenz prüfen 11-16
Regeln für Variablendeklaration..... 11-3
Regeln zum Festlegen von Bausteineigenschaften.....11-5
Regeln zum Festlegen von Systemattribute.....11-4
Regeln zur Reihenfolge der Bausteine.....11-4
speichern 11-15
Struktur von anwenderdefinierten Datentypen.....11-9
Struktur von Bausteinen.....11-8
Struktur von Codebausteinen.....11-8
Struktur von Datenbausteinen 11-9
Syntax für Bausteine.....11-10
übersetzen 11-16, 11-17
Basissoftware.....1-6
Baugruppe
parametrieren A-115, A-116
simulieren.....20-1
Baugruppen tauschen in der Konfigurationstabelle 25-1
Baugruppenadressierung.....A-110
Baugruppen-Anfangsadresse.....A-111
Baugruppenparameter.....A-115, A-116
übertragen mit SFCs.....A-115
übertragen mit STEP 7 .....A-115
Baugruppenträgerausfall (OB 86).....21-38
Baugruppentyp-abhängiger Umfang der Auskunftsfunktionen im Baugruppenzustand.....21-12
Baugruppenzustand 1-10, 21-12, 21-13
Aktualisieren.....21-11
Aufrufmöglichkeiten 21-9
Auskunftsfunktionen 21-10
Baugruppenzustand anzeigen.....21-1
Baugruppenzustand aus der Projektsicht (Online) aufrufen.....21-6
Baumstruktur..... 12-4, 12-5
Baustein
Aufrufumgebung festlegen 19-9
zum Verändern des Pointers..... A-53
Bausteinaufrufe 4-9, 4-10
aktualisieren 9-23
Bausteinbezogene Meldungen
anlegen 14-10, 14-11
zuordnen und bearbeiten 14-6
Bausteine..... 4-2, 4-3, 13-1, 13-2
Attribute 8-16
auf dem Zielsystem löschen.....17-20
aus S7-CPU zurückladen.....17-15
eingeben in AWL 9-12
erzeugen mit S7-GRAPH.....8-6
im Anwenderprogramm.....4-2
in das Zielsystem nachladen 17-5
speichern 9-24

umverdrahten 8-15
Zugriffsrechte..... 9-3
Bausteine (geladene)
im PG/PC bearbeiten 17-15
Bausteineigenschaft
Zeitstempel..... 13-3
Bausteineigenschaften..... 8-12, 8-13, 8-14, 9-1
Baustein-Eigenschaften
Baustein-Längen anzeigen..... 8-14
Bausteinkommentar..... 9-13
Bausteinkonsistenz prüfen 13-1
Baustein-Längen
anzeigen..... 8-14
Baustein-Ordner 5-12, 8-10
Bausteinordner-Eigenschaften
Baustein-Längen anzeigen..... 8-14
Baustein-Stack A-13, A-22
Bausteintitel..... 9-13
Bausteinvorlagen in AWL-Quellen
einfügen 11-14
BCD A-37
BCD-Format A-38
Bearbeiten
geladener Bausteine im PG/PC 17-15
Projekt..... 6-8
S7-Quellen 11-13
Bedienelemente beschreiben am Beispiel
eines industriellen Mischprozesses..... 3-9
Bedienen und Beobachten von Variablen ... 15-1
Bedienerrelevante Texte
exportieren/importieren.....14-16
übersetzen und bearbeiten.....14-16
Voraussetzungen.....14-16
Bedienphilosophie 5-18
Bedienpult Beschreiben 3-9
Bedienung über Tastatureingabe..... 5-30
Beispiel
Eingabe eines zusammenhängenden
Operandenbereichs..... 18-9
Eingabe von Operanden in
Variablentabellen 18-8
FBs in AWL-Quellen.....11-22
Gleitpunktzahlenformat A-36
Maskieren und Demaskieren von
Synchronfehlerereignissen A-104
Sperren und Freigeben von Alarm- und
Asynchronereignissen
(SFC 39 und 40) A-108
UDTs in AWL-Quellen.....11-24
verzögerte Bearbeitung von Alarm- und
Asynchronereignissen
(SFC 41 und 42) A-109
Beispiel zum Arbeiten mit
Verwendungsstellen 12-13
Beispiele
DBs in AWL-Quellen.....11-23
Eingabe von Steuer-/Forcewerten 18-9
Variablendeklarationen in AWL-Quellen 11-18
Beispielprogramm
Industrieller Mischprozeß A-75
Beispielprogramme A-74
Beispielprojekte..... A-75
Belegungsplan
Ausgänge und Merker (E/A/M) 12-7
Zeiten und Zähler (T/Z) 12-7
Benutzungsoberfläche.....5-19
Beobachten
Prinzipielle Vorgehensweise.....18-2
Beobachten von Variablen
Einführung.....18-13
Trigger festlegen.....18-13
Beschreiben
der einzelnen Funktionsbereiche.....3-4
der notwendigen Anzeige- und
Bedienelemente.....3-9
Sicherheitsanforderungen für das Beispiel
eines industriellen Mischprozesses.....3-7
Beschreiben des Bedienpultes am Beispiel
eines industriellen Mischprozesses.....3-9
Beschreibung der einzelnen Aufgaben und
Bereiche für das Beispiel eines
industriellen Mischprozesses.....3-4
Besonderheiten zum Drucken des
Objektbaums.....22-4
Bestandteile einer Meldung.....14-4
Betriebsstundenzähler A-117, A-118
Betriebssystem
Aufgaben.....4-1
Betriebssystem der CPU.....4-15
Betriebssysteme für M7-300/400.....24-5
Betriebszustand
ANLAUF.....A-1, A-3, A-4, A-5
anzeigen und ändern 16-4
HALT A-1, A-4
Wissenswertes 19-7
RUN.....A-1, A-3, A-4
STOP.....A-1, A-2, A-3, A-4
Stack-Inhalte 21-14
Betriebszustand "RUN"..... A-11
Betriebszustand der CPU beim Laden
ändern.....17-8
Betriebszustand HALT..... A-12
Betriebszustände
der CPU A-1
Priorität A-4

Betriebszustände und Übergänge A-1
Betriebszustandsübergänge A-2
Bibliothek 5-7
Bibliotheken 6-7
  arbeiten mit 8-17
  archivieren 22-5
  hierarchischer Aufbau 8-19
  reorganisieren 25-2
Binär-codierte Dezimalzahlen (BCD) A-37
Bitmeldeverfahren 14-1, 14-2
BLKMOV A-15
BLOCK
  Parametertyp A-49
BLOCK_DB A-50
BLOCK_FB A-50
BLOCK_FC A-50
BLOCK_SDB A-50
BOOL
  als Datentyp A-29
  Bereich A-30
Boxen
  entfernen
  ändern 9-21
  plazieren 9-16, 9-20
Browser 5-26
B-Stack
  im B-Stack gespeicherte Daten A-22
  verschachtelte Aufrufe A-22
BuB-Attribute 15-1
  ändern mit CFC 15-4
  projektieren mit AWL
  FUP
  KOP 15-2
  projektieren über die Symboltabelle 15-3
Byte
  als Datentyp A-29
  Bereich A-30
CAN_TINT 4-26
CFC 8-3
CFC-Programm 24-1
Codebausteine
  Aufbau 9-1
  definieren
  Beispiel A-78
  im inkrementellen Editor 9-1
  speichern 9-24
  Zeitstempel 13-4
COUNTER A-49
  Parametertyp A-49
CPU (Central Processing Unit)
  Betriebszustände A-1, A-2, A-3, A-4
CPU laden mit einer Konfiguration 17-7
CPU urlöschen 17-19
CPU-Baugruppe simulieren 20-1
CPU-Hardwarefehler (OB 84) 21-37
CPU-Hardwarefehler-OB 21-37
CPU-Meldungen
  anzeigen 14-20
  Archivgröße 14-21
CPU-Meldungen konfigurieren 14-22
CPU-Meldungen und Anwenderdefinierte
  Diagnosemeldungen anzeigen 14-20
CPU-Parameter "Zyklusbelastung durch
  Kommunikation" 4-15
CPU-Redundanzfehler (OB 72) 21-31
CREAT_DB A-14
CRST/WRST A-5, A-6, A-7
CTRL_RTM A-117
Darstellung
  FUP-Elemente 9-19
  globale oder lokale Symbole 7-4
  KOP-Elemente 9-15
Das STEP7 - Basispaket 1-6
DATE AND TIME (Datum und Uhrzeit)
  Bereich A-41
  Format A-40
DATE_AND_TIME A-39
Dateiformate für den Import/Export einer
  Symboltabelle 7-16
Datenaustausch
  in verschiedenen Betriebszuständen A-12
Datenbaustein (DB) 4-2
  global 4-22
  Instanz-Datenbausteine 4-17, 4-20
  remanent A-26
  Struktur 4-22
Datenbausteine
  Datensicht 10-3
  Datenstruktur mit zugeordnetem FB (Instanz-
    DBs) eingeben/anzeigen 10-5
  Datenwerte auf die Anfangswerte rücksetzen
    10-8
  Datenwerte in der Datensicht ändern 10-8
  Deklarationssicht 10-2
  Grundlagen 10-1
  speichern 10-9
Datenbausteinregister A-23
Datensatz
  lesen A-112
  schreiben A-112
  zugreifen auf A-112, A-115
Datensicht von Datenbausteinen 10-3
Datentyp
  anwenderdefiniert 8-11
  DATE AND TIME: Datum und Uhrzeit A-40
  DWORD A-37

REAL:Gleitpunktzahl A-33, A-34
S5 TIME A-38
UDT 8-11
WORD A-37
Datentypen
anwenderdefiniert A-39
ANY A-56
ARRAY A-39
Beschreibung A-30
BOOL A-29, A-30
Byte A-29
BYTE A-30
DATE_AND_TIME A-39
Datum A-30
DINT: Ganzzahl (32 Bit) A-32
Doppelwort A-29
Doppelwort (DWORD) A-30
Einführung A-29
elementar A-30
FB, SFB 4-17, A-39
Ganzzahl (16 Bit) (INT) A-30
Ganzzahl (32 Bit) (DINT) A-30
INT: Ganzzahl (16 Bit) A-31
Parametertypen, ANY, Parameter A-58, A-59, A-60
Realzahl (REAL) A-30
S5 TIME A-30
STRING A-39
STRUCT A-39
UDT A-39
Uhrzeit (TIME OF DAY) A-30
Wort A-29
Wort (WORD) A-30
Zeichen (CHAR) A-30
Zeit (TIME) A-30
zusammengesetzt A-39
Datentypen zu Lokaldaten von Codebausteinen zuordnen A-62
Datenwerte
auf die Anfangswerte rücksetzen 10-8
in der Datensicht von Datenbausteinen ändern 10-8
Datenzykluszeit 1-10
DB 4-22
Formattabelle 11-12
DBs in AWL-Quellen
Beispiele 11-23
Deaktivieren
Uhrzeitalarm 4-25
Dearchivieren
Vorgehensweise 22-6
Defekt Betriebszustand der CPU A-1
Definieren
Codebausteine A-78
Symbole bei der Programmeingabe 7-11
Deinstallieren
Autorisierung 2-1, 2-2, 2-3, 2-4, 2-5
STEP 7 2-14
Deklarationssicht von Datenbausteinen 10-2
Deklarationstabelle
Multiinstanz eingeben 9-10
Deklarationstyp
ändern 9-7, 9-8
Deklarieren von lokalen Variablen
FB für das Beispiel eines industriellen Mischprozesses A-81
OB für das Beispiel eines industriellen Mischprozesses A-87
Deklarieren von Parametern
FC für das Beispiel eines industriellen Mischprozesses A-85
Demaskieren
Startereignisse 4-33
Demaskieren von Synchronfehlerereignissen
Beispiel A-104
Dezentrale Peripherie A-72
Diagnose der Hardware 21-1
Diagnosealarm (OB 82) 21-35
Diagnosealarm-OB 21-35, 21-37
Diagnosedaten auf Baugruppen 21-20
Diagnoseereignis 21-21
Diagnosefunktionen 21-21
Diagnosemeldung
an Teilnehmer senden 21-20
eigene schreiben 21-20
Diagnosepuffer A-24, A-25
auslesen 21-17, A-25
auswerten A-24
Definition A-24
Inhalt 21-21, A-24, A-25
Diagnosesymbole
in der Online-Sicht 21-3
Diagnosezustandsdaten 21-19
Dialogfelder 5-21
DINT
Datentyp A-32
DIS_AIRT 4-35
DIS_IRT 4-35
DMSK_FLT 4-35
DOCPRO 22-2
Dokumentation 1-1, 5-5, 5-6, 22-1
eines ganzen Projekts drucken 22-1
von Projektbestandteilen drucken 22-1
Doppelwort (DWORD)
Bereich A-30
Datentyp A-29

Download (Netzkonfiguration) 17-10
DP-Mastersysteme 1-10
DP-Normslaves A-114
DPNRM_DG..... A-114
DPRD_DAT..... A-114
DP-Slaves A-72
DP-Slaves mit fehlenden oder fehlerhaften
GSD-Dateien A-73
DPWR_DAT A-114
Drucken
Bausteine 22-1
Diagnosepufferinhalt 22-1
Globaldatentabelle 22-1
Konfigurationstabelle..... 22-1
Projektbestandteile..... 22-1
Projektdokumentation 22-1
Referenzdaten 22-1
Symboltabelle 22-1
Variablentabelle 22-1
Drucker einrichten..... 22-4
Dummy-Slave..... A-73
Durchgänge auflisten..... 3-6
DWORD Datentyp A-37
Editieren in der Symboltabelle.....7-11, 7-12
Editor
Einstellungen für AWL..... 9-2
Eigene Diagnosemeldungen senden 21-20
Einfügen
Bausteinvorlagen in AWL-Quellen 11-14
Ersatzwerte bei Fehlererkennung 21-29
externer Quellen 11-14
Kommentarzeilen..... 18-7
Operanden oder Symbole in eine
Variablentabelle 18-4
S7-/M7-Programm 6-6
Steuerwerte 18-5
Einfügen von Stationen..... 6-4
Einführung zu Daten- und Parametertypen . A-29
Einführung zum Forcen von Variablen.....18-18
Einführung zum Testen mit der
Variablentabelle 18-1
Eingabe in Dialogfelder..... 5-19
Eingabemöglichkeiten von globalen
Symbolen 7-10
Eingänge
auflisten..... 3-6
Belegungsplan..... 12-6
Prozeßabbild A-18
Eingangsdiagramm für die Motoren erstellen 3-6
Eingangsdiagramm für die Ventile erstellen .. 3-7
Eingeben
Datenstruktur von Datenbausteinen mit
zugeordnetem FB (Instanz-DBs)..... 10-5
Datenstruktur von globalen
Datenbausteinen.....10-4
einzelner globaler Symbole im Dialog.....7-11
globale Symbole in ein Programm.....9-12
Multiinstanz in der
Variablendeklarationstabelle.....9-10
Struktur von anwenderdefinierten
Datentypen (UDT).....10-6
Struktur von Datenbausteinen mit
zugeordnetem UDT.....10-7
von Symbolen.....7-12
Eingeben mehrerer globaler Symbole in
der Symboltabelle.....7-12
Eingrenzen von Störungen.....21-2
Einsetzen
SFC A-19
Einstellen
Aufrufumgebung des Bausteins.....19-9
Betriebsverhalten.....A-114
Operandenvorrang 7-5
PG/PC-Schnittstelle.....2-12
Uhrzeit und Datum.....16-5
virtueller Arbeitsspeicher.....25-5
Einstellungen
AWL-Editor.....9-2
für die Programmiersprache AWL.....9-22
für die Programmiersprache FUP.....9-19
Einstellungen für das Melden von
Systemfehlern 14-26
Einstellungen für Programmiersprache KOP9-15
Elementare Datentypen A-30
Elemente in Dialogfeldern.....5-19
EN / ENO
Beschaltung 9-21
EN_AIRT 4-35
EN_IRT.....4-35
Engineering Tools 1-15
Entstehung von Lücken im Anwenderspeicher
(RAM) 17-20
EPROM A-26
EPROM-Bereich..... A-14
Ereignisse.....4-15, 4-16
Ereignisse
asynchron 4-11
Erreichbare Teilnehmer 16-1
Erreichbare Teilnehmer anzeigen.....16-1
Ersatzwert
Verwenden von SFC44 (RPL_VAL).....21-29
Erstellen
eines anwenderdefinierten Datentyps..... A-47
Eingangs-/Ausgangs-Diagramm für die
Motoren.....3-6

Eingangs-/Ausgangs-Diagramm
für die Ventile ... 3-7
FB für den Motor ... A-81, A-82, A-83
FC für die Ventile ... A-85, A-86
Felder ... A-42, A-44, A-45
Konfigurationsplan ... 3-10
OB 1 für das Beispiel eines industriellen Mischprozesses ... A-87
Struktur ... A-45, A-46
Variablentabelle ... 18-2
von Anwenderprogrammen ... 9-1
Erstellen eines Konfigurationsplans am Beispiel industrieller Mischprozeß ... 3-10
Erstmaliges Laden der Netzkonfiguration ... 17-8
Erweitern von DP-Slaves (mit Vorgänger-Versionen von STEP 7 erstellt) ... A-72
Erweiterungsmöglichkeiten des STEP7 - Basispakets ... 1-14
Erzeugen ... 5-21, 5-22
Objekte ... 5-21
Referenzdaten ... 12-11
und Manipulieren von Objekten ... 5-21
Erzeugen und Manipulieren von Objekten ... 5-21
Erzeugte Fehler-OBs (Systemfehler melden) ... 14-28
Exportieren Symboltabelle ... 7-15
Externe Quellen einfügen ... 11-14
FB ... 4-17, 4-18, 4-19, A-39
Formattabelle ... 11-11
Korrigieren der Schnittstelle ... 9-24
FBs in AWL-Quellen
Beispiel ... 11-22
FC ... 4-16, 4-17
Formattabelle ... 11-11
Korrigieren der Schnittstelle ... 9-24
FC 12 ... A-94
FCs in AWL-Quellen
Beispiel ... 11-20, 11-21
Fehlende Symbole anzeigen ... 12-10
Fehler während der Installation ... 2-9
Fehler vermeiden beim Aufruf von Bausteinen ... 9-25
Fehlerbehebung
Programmbeispiele ... 21-24
Fehlererkennung
OB-Typen
OB81 ... 21-24
Programmbeispiele
Ersatzwerte ... 21-29
Verwenden von Fehler-OBs
um auf Fehler zu reagieren ... 4-33
Fehler-OB ... 21-24, 21-25, 21-26, 21-27, 21-28
OB-Typen
OB121 und OB122 ... 4-33
OB70 und OB72 ... 4-33
OB80 bis OB87 ... 4-33
Verwenden von Fehler-OBs zur Reaktion auf Ereignisse ... 4-33
Fehler-OBs ... 14-28
Fehler-OBs als Reaktion auf die Erkennung eines Fehlers ... 21-24
Fehlersuche ... 21-1
in Bausteinen ... 9-14
Fehlersuche in AWL-Quellen ... 11-16
Feld (Datentyp ARRAY)
Anzahl verschachtelter Ebenen ... A-41
Beschreibung ... A-42
Felder erstellen ... A-42, A-44
für den Datenzugriff verwenden ... A-42
Fensteranordnung wiederherstellen ... 5-29
Fenstertypen umschalten ... 5-34
FEPROM ... A-26
Festlegen
Sicherheitsanforderungen ... 3-7
Trigger zum Beobachten von Variablen ... 18-13
Trigger zum Steuern von Variablen ... 18-16
Feststellbare Fehler ... 21-24
Filtern von Symbolen ... 7-12
Flash-File-System ... 2-9
Forcen von Variablen ... 18-18
Einführung ... 18-18
Sicherheitsvorkehrungen ... 18-20
Forcewerte
Beispiele für die Eingabe ... 18-9
Formalparameter
Systemattribute und Meldebausteine ... 14-7
Format
BLOCK ... A-50
COUNTER ... A-50
Datentyp DATE_AND_TIME (Datum und Uhrzeit) ... A-40
Datentyp DINT (32-Bit-Ganzzahlen) ... A-32
Datentyp INT (16-Bit-Ganzzahlen) ... A-31
Datentyp REAL (Gleitpunktzahlen) ... A-32
Datentyp S5TIME (Zeitdauer) ... A-38
Datentypen WORD und DWORD bei binär-codierten Dezimalzahlen ... A-37
Parametertyp ANY ... A-56
Parametertyp POINTER ... A-51
Parametertypen BLOCK
COUNTER
TIMER ... A-50
TIMER ... A-50
Formate für Bausteine in AWL-Quellen ... 11-10
Formattabelle von DBs ... 11-12

Formattabelle von FBs.....11-11
Formattabelle von FCs.....11-11
Formattabelle von OBs.....11-10
Freigeben von Alarm- und
Asynchronereignissen
Beispiel..... A-108
Funktion (FC).....4-2, 4-16
Anwendungsbereich..... 4-17
Erstellen
Beispiel-FC für das Beispiel eines
industriellen Mischprozesses..... A-85
Funktionen..... 22-2
Funktionen (FC)..... 4-16
Funktionsbausteine (FB)..... 4-17
Funktionsbausteine (FBs).....4-2, 4-17
Aktualparameter.....4-18, 4-19
Anwendungsbereich..... 4-17
Erstellen
Beispiel-FB für das Beispiel eines industriellen
Mischprozesses..... A-81
Funktionsleiste Symbole..... 5-19
Funktionsplan..... 8-4
Funktionsumfang von
"Systemfehler melden".....14-24
FUP..... 8-4
Anzeige sprachabhängiger Information... 12-9
FUP-Anweisungen
Regeln für die Eingabe..... 9-20
FUP-Elemente
Darstellung..... 9-19
FUP-Layout..... 9-19
Ganzzahl (16 Bit)
Format..... A-31
Ganzzahl (16 Bit) (INT)
Bereich..... A-30
Ganzzahl (32 Bit)
Format..... A-32
Ganzzahl (32 Bit) (DINT)
Bereich..... A-30
GD-Kommunikation..... A-72
Geladene Bausteine im PG/PC bearbeiten 17-15
Generieren
AWL-Quellen aus Bausteinen.....11-15
Gepufferter Betrieb
remanente Speicherbereiche..... A-27
Gleitpunktzahl
Beispiel..... A-36
Grundelemente..... A-32
Komponentenfelder..... A-32
Parameter..... A-32
Gleitpunktzahlen
Format..... A-33, A-34
Globaldaten-Kommunikation..... A-72
Globaldaten-Konfiguration laden.....17-12
Globale Datenbausteine
Datenstruktur eingeben.....10-4
Zeitstempel.....13-5
Globale Datenbausteine (DB).....4-22
Globale Symbole
einzeln eingeben im Dialog.....7-11
in ein Programm eingeben.....9-12
mehrere eingeben in der Symboltabelle...7-12
Globale und lokale Symbole.....7-3
GRAPH.....8-3
Groß-/Kleinschreibung bei Symbolen.....7-13
Große Projekte.....25-1
Grundlagen
zu Datenbausteinen.....10-1
Grundlagen des Meldekonzeptes.....14-1
Grundlagen zum Programmieren in
AWL-Quellen.....11-1
GSD-Datei..... A-72, A-73
HALT..... A-12
Betriebszustand der CPU..... A-1
Haltepunktleiste.....19-6
Hantierung von großen Projekten.....25-1
Hardware diagnostizieren.....21-1
Ausführliche Diagnosesicht.....21-8
Schnellansicht.....21-5
Hardware Katalog.....1-10
Herstellen einer Verbindung zur CPU.....18-12
Hierarchischer Aufbau von Bibliotheken.....8-19
HiGraph.....8-3
Hilfe (online)
aufrufen.....5-4
Themen.....5-4
Hintergrund-OB
Priorität.....4-32
Programmieren.....4-33
Hinweis
zur Aktualisierung des Fensterinhalts.....16-4
Hochrüsten Autorisierung.....2-1
Human Machine Interface.....1-18
Icons für Objekte im SIMATIC Manager.....5-5
Ident-Nummer eintragen.....2-9
Importieren externe Quelle.....6-6
Importieren Symboltabelle.....7-15
Indirekte Parametrierung.....A-115
Installationsvoraussetzungen.....2-8
Installieren
STEP 7.....2-8, 2-9
Installieren der Autorisierung nach
dem Setup.....2-1
Installieren der Autorisierung während des
Setups.....2-1

Installieren und Deinstallieren der Autorisierung 2-1
Installieren von STEP 7..... 2-8
Instanz 4-20
Instanz-Datenbaustein 4-20, 4-21
mehrere Instanzen für einen FB erstellen 4-17
remanent A-26
Instanz-Datenbausteine 4-20
Zeitstempel..... 13-6
INT
Datentyp..... A-31
Interruptbelegung prüfen..... 2-13
Kaltstart..... A-1
Kennzeichnung von Symbolen..... 7-4
Kombinationsfeld
Definition 5-19
Kommentare
zu Bausteinen..... 9-13
zu Netzwerken 9-13
Kommentarzeichen..... 18-4
Kommentarzeile..... 18-4
Kommentarzeilen einfügen..... 18-7
Kommunikation.....17-8, 17-10
Kommunikationsfehler (OB 87).....21-39
Kommunikationsfehler-OB 21-39
Kommunikationslast.....4-15, 4-16
Kommunikationsprozesse 4-11
Kommunikations-Redundanzfehler OB 73 .21-32
Kompatibilität (DP-Slaves) A-72, A-73
Komprimieren
des Anwenderspeichers 17-20
des Speicherinhalts einer S7-CPU.....17-21
Komprimieren des Anwenderspeichers (RAM)
nach mehrfachen Löschen oder Nachladen 17-20
Konfiguration laden in das PG.....17-16
Konfigurierbare Speicherobjekte im Arbeitsspeicher..... A-28
Konfigurieren..... 25-1
Konfigurieren von CPU-Meldungen 14-22
Konfiurationsplan erstellen 3-10
Konsistenz einer Stationskonfiguration prüfen..... 17-7
Konsistenz in AWL-Quellen prüfen 11-16
Kontaktplan 8-4
Kontextabhängige Hilfe..... 5-4
Kontrolle der Zykluszeiten zur Vermeidung von Zeitfehlern.....21-16
Konzeption einer Automatisierungslösung
Anzeige- und Bedienelemente beschreiben 3-9
Ein-
Aus- und Durchgänge auflisten 3-6
Eingangs-/ Ausgangs-Diagramm für die Ventile erstellen.....3-7
Eingangs-/ Ausgangs-Diagramms für die Motoren erstellen.....3-6
Funktionsbereiche beschreiben.....3-4
Konfigurationsplan erstellen.....3-10
Prinzipielle Vorgehensweise.....3-1
Prozeß in Aufgaben und Bereiche zerlegen 3-2
Sicherheitsanforderungen festlegen 3-7
KOP..... 8-2, 8-3, 8-4
Anzeige sprachabhängiger Information....12-9
unzulässige Verschaltungen.....9-18
KOP-Anweisungen
Regeln für die Eingabe 9-16
KOP-Elemente
Darstellung.....9-15
Kopf- und Fußzeilen 22-3
Kopieren/Verschieben von Variablentabellen.....18-3
Kopierschutz 2-1
KOP-Layout 9-15
Korrigieren
der Schnittstellen in einem FC FB oder UDT 9-24
Kurzschluß.....9-18
Lade- und Arbeitsspeicher A-14
Lade- und Arbeitsspeicher in der CPU 17-3
Lade-/Arbeitsspeicher löschen.....17-19
Lademöglichkeiten A-14
Lademöglichkeiten abhängig vom Ladespeicher.....17-4
Laden
Anwenderprogramm A-14
Anwenderprogramme in Zielsystem 17-3
aus dem Zielsystem in das PG.....17-14
der aktuellen Konfiguration und aller Bausteine in das PG 17-14
über EPROM-Memory Cards 17-6
Voraussetzungen.....17-1
Laden der Globaldaten-Konfiguration.....17-12
Laden der Netzkonfiguration in ein Zielsystem 17-10
Laden einer Konfiguration in das PG.....17-16
Laden einer Konfiguration in ein Zielsystem 17-7
Laden von Änderungen der Netzkonfiguration 17-10
Ladespeicher 17-3, A-13, A-14
nicht ablaufrelevante DBs speichern..... A-14
Leitfaden durch STEP~7.....1-1
Leittechnikdialog

siehe PCS7-Meldungsprojektierung.....14-12
Linear programmieren..... 4-3
Listen bedienerrelevanter Texte 14-16
Listenfeld..... 5-19
Lokaldatenbedarf.....12-4, 12-5
Lokaldaten-Stack..... A-13, A-22
Löschen
Lade-/Arbeitsspeicher 17-19
S7-Bausteine auf dem Zielsystem 17-20
STEP 7-Objekte..... 5-21
L-Stack
Bearbeitung von Daten in einem verschachtelten Aufruf..... A-22
Speichern von temporären Variablen..... 4-17
Überschreiben des L-Stack A-21
Zuordnen von Speicher zu lokalen Variablen A-21
M7-Programm einfügen 6-6, 6-7
Make (siehe Bausteinkonsistenz prüfen)..... 13-1
Manipulieren
Objekte..... 5-21, 5-22, 5-23, 5-24, 5-25, 5-26
Maskieren Startereignisse..... 4-33
Maskieren von Synchronfehlerereignissen
Beispiel..... A-104
Maßnahmen im Programm zur Störungsbehandlung.....21-22
Maximalzykluszeit..... 4-13
Mehrbenutzer-Konfiguration..... 23-1
Mehrbenutzer-Konfiguration im Windows-Netzwerkverbund..... 23-1
Meldebausteine
Übersicht..... 14-6
Meldefähiger Baustein 14-9
Meldekonzept
Grundlagen..... 14-1
Melden von Systemfehlern.....14-29
Melden von Systemfehlern projektieren 14-23
Meldenummern
vergeben 14-5
Meldenummernverfahren 14-2
Meldeverfahren
auswählen 14-2
Meldung
anwenderdefinierte Diagnosemeldung...14-15
bausteinbezogen..... 14-6
Beispiel..... 14-5
Bestandteile..... 14-4
symbolbezogen.....14-14
Meldungsprojektierung
Daten nach WinCC transferieren 14-19
SIMATIC-Komponenten 14-4
Meldungstyp.....14-8, 14-9
Meldungstyp und Meldungen 14-8
Memory Card A-15
parametrieren.....2-11
Merker
Belegungsplan.....12-6
Speicherbereich
remanent..... A-27
Mindestzykluszeit 4-13
Mnemonik
einstellen.....9-22
Motoren
Eingangs-/Ausgangs-Diagramm erstellen..3-6
MPI-ISA-Card (Auto) 2-12
MPI-Karte im PG/PC 2-12
MPI-Schnittstelle 2-8, 2-9
MSK_FLT.....4-35
Multiinstanz.....4-17, 4-20
in der Variablendeklarationstabelle eingeben 9-10
Multiinstanzen
Regeln 9-10
Verwendung 9-9
Multiuser-Konfiguration
siehe Mehrbenutzerkonfiguration 23-1
Nachladen von Bausteinen in das Zielsystem.....17-5
Namenskonventionen
für Projektierungsdaten.....15-1
Netzausfall..... A-6, A-7
Netzkonfiguratiaon in ein Zielsystem laden 17-10
Netzkonfiguration laden 17-8, 17-10
Netzwerke.....8-4
Abschluß in KOP 9-16
Netzwerkkommentar.....9-13
Netzwerk-Server.....23-1
Netzwerktitel 9-13, 9-14
Neustart..... A-1, A-5, A-6, A-7, A-8, A-9
abbrechen..... A-9
automatisch..... A-5
ungepuffert..... A-5
manuell A-5
Nicht verwendete Operanden anzeigen.....12-10
Nicht verwendete Symbole 12-8
Nichtflüchtiger RAM..... A-26
Notautorisierung.....2-2
Nullpointer..... A-57
Nutzdaten A-111
Nutzen der Uhrzeitfunktionen.....A-117
Nutzungsberechtigung.....2-1
NVRAM..... A-26, A-27
OB.....4-3, 4-4, 4-5, 4-6, 4-7, 4-8
Formattabelle 11-10
OB 1 .....A-102, A-103
OB 1 und OB 80..... A-98

OB 10.....A-96, A-97
OB 100.....A-5
OB 101.....A-5, A-11
OB 102.....A-5
OB 1-Zykluszeit 4-15
OB 20..... A-101, A-102
OB 70.....21-31
OB 72.....21-31, 21-32
OB 73.....21-32, 21-33
OB 80.....21-33
OB 81.....21-34
OB 82.....21-35
OB 83.....21-36
OB 84.....21-37
OB 85.....21-37, 21-38, A-20, A-21
OB 86
Baugruppenträgerausfall 21-38
OB 87.....21-39
Obergrenzen für die Eingabe von Zählern .. 18-7
Obergrenzen für die Eingabe von Zeiten..... 18-6
Objekt
ausschneiden
kopieren
einfügen 5-21, 5-22, 5-23
auswählen 5-26
Eigenschaften.....5-22, 5-23
Löschen..... 5-26
öffnen 5-22, 5-25, 5-26
umbenennen.....5-24, 5-25
verschieben 5-25
Objekt Baustein-Ordner 5-12
Objekt Bibliothek..... 5-7
Objekt erzeugen 5-21
Objekt Programmierbare Baugruppe 5-9
Objekt Projekt..... 5-6
Objekt Quellen-Ordner..... 5-15
Objekt S7/M7-Programm 5-11
Objekt Station..... 5-8
Objekte.....5-5, 5-6
als Ordner..... 5-5
als Träger von Eigenschaften 5-5
alsTräger von Funktionen..... 5-5
manipulieren..... 5-21
Objekte und Objekt-Hierarchie 5-5
Objekthierarchie 5-5
aufbauen 5-22
OBs in AWL-Quellen
Beispiel.....11-19
Öffnen Variablentabelle.....18-2, 18-3
von Symboltabellen..... 7-12
Online-Hilfe aufrufen..... 5-4
Schriftgröße ändern 5-4
Themen..... 5-4
Online-Sicht
Diagnosesymbole 21-3
Online-Verbindung
aufbauen über das Fenster "Erreichbare Teilnehmer" 16-1
aufbauen über das Online-Fenster des Projekts 16-2
Operanden
in eine Variablentabelle einfügen.....18-4
nicht verwendete.....12-11
ohne Symbol 12-11
umverdrahten 8-15, 8-16
Operanden ohne Symbol.....12-9
Operandenbereiche
Beschreibung A-16
Operandenfeldbreite..... 9-15, 9-19
Operandenvorrang
einstellen.....7-5
Optionspaket.....20-1
Optionssoftware 24-3
Optionssoftware für M7-Programmierung 24-3
Ordner
Bausteine 8-10
Organisationsbaustein (OB)
Hintergrund-OB (OB 90) 4-3, 4-32
Organisationsbaustein für die Programmbearbeitung im Hintergrund (OB 90) 4-32
Organisationsbaustein für zyklische Programmbearbeitung (OB1) 4-11
Organisationsbausteine 4-3
Definition.....4-3
Erstellen eines OBs für das Beispiel eines industriellen Mischprozesses..... A-87
Ersatzwerte 21-29
Prioritätsklassen 4-4, 4-6, 4-7
Reagieren auf Fehler 4-33
Startinformation.....4-6
Organisationsbausteine für alarmgesteuerte Programmbearbeitung 4-24
Organisationsbausteine für den Anlauf (OB100/OB101/OB102) 4-30
Organisationsbausteine für den Prozeßalarm(OB 40 bis OB 47).....4-29
Organisationsbausteine für den Uhrzeitalarm (OB 10 bis OB 17) 4-25
Organisationsbausteine für den Verzögerungsalarm (OB 20 bis OB 23)....4-27
Organisationsbausteine für den Weckalarm (OB 30 bis OB 38) 4-27
Organisationsbausteine für die Fehlerbearbeitung (OB70bisOB87 / OB 121 bis OB 122) 4-33

Organisationsbausteine und
Programmstruktur 4-3
Outdoor 1-10
Parameter Attribute 8-16
Parametertyp
ANY A-49
BLOCK_DB A-49
BLOCK_FB A-49
BLOCK_FC A-49
BLOCK_SDB A-49
COUNTER A-50
POINTER A-49
TIMER A-49
Parametertyp POINTER verwenden A-52
Parametertypen A-49
ANY A-58, A-59, A-60, A-61
COUNTER A-49
Einführung A-29
Parametrierbare Baugruppen A-115
Parametrieren
indirekt A-115
mit SFCs A-116
mit STEP 7 ..... A-115, A-116
prozeßalarmfähige Signalbaugruppen 4-29
Uhr A-117
Parametrieren der PG/PC-Schnittstelle 2-12
PARM_MOD A-112, A-116
Paßwort 16-3
Paßwortschutz für Zugriff auf Zielsysteme .. 16-3
PCS7-Meldungsprojektierung 14-12, 14-13
PC-Station 1-10
Peripherie Adreßbereiche A-110, A-111
Peripheriedaten A-111
Peripherie-Redundanzfehler (OB 70) 21-31
Peripherie-Redundanzfehler-OB 21-31
Peripheriezugriffsfehler (OB 122) 21-40
Peripheriezugriffsfehler (PZF) bei
Prozeßabbild-Aktualisierung A-20
Peripheriezugriffsfehler-OB 21-40
Personenschaden verhüten 18-18
PG/PC-Schnittstelle 2-12, 2-13
Parametrieren 2-12
Phasenverschiebung 4-28
Plazieren Boxen 9-20
POINTER A-50
Parametertyp A-49
Pointerformat A-49, A-51
Prinzipielle Vorgehensweise
beim Beobachten und Steuern 18-2
beim Drucken 22-2
beim Erstellen von Codebausteinen 9-1
zur Konzeption einer
Automatisierungslösung 3-1
Prinzipielles Vorgehen
zum Ermitteln einer STOP-Ursache 21-14
Priorität
ändern 4-6
Hintergrund-OB 4-32
Prozeßalarm 4-30
Uhrzeitalarm 4-25
Verzögerungsalarm 4-27
PROFIBUS-DP 1-10, A-72
Profile 1-10
Programmablauffehler (OB 85) 21-37
Programmablauffehler-OB 21-37
Programmbearbeitung
alarmgesteuert 4-3, 4-24
zyklisch 4-8
Programmbeispiele
Beispiel eines industriellen Mischprozesses
Beschreiben der Sicherheitsanforderungen 3-7
Beschreiben des Bedienpultes 3-9
Beschreibung der einzelnen Aufgaben und
Bereiche 3-4
Erstellen eines E/A-Diagramms 3-6
Konfigurationsplan erstellen 3-10
Zerlegen eines Prozesses in
Aufgabenbereiche 3-2
Einfügen von Ersatzwerten 21-29
Ersatzwerte 21-29
FB für das Beispiel eines industriellen
Mischprozesses A-81
FC für das Beispiel eines industriellen
Mischprozesses A-85
OB für das Beispiel eines industriellen
Mischprozesses A-87
Reagieren auf Batteriefehler 21-24
Programme in einer CPU 4-1
Programmelemente
einfügen 9-4
Programmelemente-Katalog 9-4
Programmerstellmethode auswählen 8-1
Programmerstellung
allgemeine Vorgehensweise 1-1
Programmieren einer FC Beispiel A-85
eines OB 1 Beispiel A-87
FB Beispiel A-81
Hintergrund-OB 4-32
Programmierfehler (OB 121) 21-39
Programmierfehler-OB 21-39
Programmierschritte S7 1-1
Programmiersprache
auswählen 8-2
festlegen 8-2
FUP (Funktionsplan) 8-4
KOP (Kontaktplan) 8-4

S7-CFC 8-9
S7-Graph (Ablaufsteuerung)..... 8-6
S7-HiGraph (Zustandsgraph)..... 8-7
S7-SCL..... 8-5
Programmiersprache AWL (Anweisungsliste)..... 8-5
Programmiersprachen.....1-6, 1-8
Programmierung
Entwerfen eines strukturierten Programms..... A-78
Übergeben von Parametern 4-17
Verwendung von Datenbausteinen 4-17
Programmstatus 19-1, 19-9
anzeigen..... 19-3
bei Multiinstanzaufrufen..... 19-9
Programmstatus von Datenbausteinen 19-8
Programmstruktur.....12-4, 12-5
anzeigen.....12-10
Projekt..... 5-6
anlegen manuell 6-3
anlegen über Assistenten 6-3
kopieren 6-8
löschen..... 6-8
mit Globaldaten-Kommunikation archivieren
umsetzen umbenennen..... A-72
öffnen..... 6-8
Projektbestandteile drucken 22-1
Projektdokumentation drucken 22-1
Projekte
archivieren..... 22-4
Bearbeitungsreihenfolge..... 6-4
reorganisieren..... 25-2
umbenennen..... 5-21
Projekte mit einer großen Anzahl vernetzter Stationen..... 25-1
Projektfenster 6-1, 6-2
Projektieren von bedien- und beobachtbaren Variablen..... 15-1
Projektieren von BuB-Attributen mit AWL KOP und FUP..... 15-2
Projektieren von BuB-Attributen über die Symboltabelle..... 15-3
Projektieren von Meldungen bei Systemfehlern 14-23
Projektierungsdaten.....15-1, 15-2
transferieren 14-19, 15-5
Voraussetzungen zum Transferieren 14-19, 15-5
Projektierungsdaten zum Zielsystem transferieren 14-19
Projektsicht..... 6-2
Projektstruktur 6-2
Prozeß
untergliedern 3-2, A-75
Prozeß zerlegen in Aufgaben und Bereiche...3-2
Prozeß zerlegen in Aufgabenbereiche für das Beispiel eines industriellen Mischprozesses.....3-2
Prozeßabbild.....4-11, A-18
aktualisieren 4-15
Ein-/Ausgänge..... A-19
löschen 4-31
Prozeßalarm
Priorität 4-30
Regeln 4-29
starten.....4-29
Prozeßalarme 4-29, 4-30
Prozeßalarmfähige Signalbaugruppen parametrieren.....4-29
Prozeßbetrieb 4-15, 4-16
Prozeßüberwachung 15-1, 18-2
Prüfen
Konsistenz in AWL-Quellen.....11-16
Referenzdaten.....12-11
Prüfen der Konsistenz einer Stationskonfiguration 17-7
PZF (Peripheriezugriffsfehler) A-18
QRY_TINT 4-25
Quelldateien in S7-GRAPH.....8-6
Quellen
externe.....6-6
Zugriffsrechte 9-3
Quellen (externe) einfügen.....11-14
Quellen-Ordner 5-15, 5-16
Querverweisliste.....12-2
RAM A-13, A-26
RAM-Bereich..... A-14, A-27
RDSYSST..... 21-17, 21-19, A-25
READ_CLK.....A-117
READ_RTM A-118
Real
Datentyp A-32
Realzahl
Bereich..... A-30
Datentyp A-30
Referenzdaten..... 12-1, 12-2
Anwendung 12-1
anzeigen 12-10, 12-11
erzeugen.....12-11
Regeln
für den Umgang mit Autorisierungen 2-5
für die Bildung von Multiinstanzen 9-10
für die Eingabe von AWL-Anweisungen...9-22
für die Eingabe von FUP-Anweisungen ...9-20
für die Eingabe von KOP-Anweisungen...9-16
für FUP 9-20

für KOP ... 9-16
für Variablendeklaration in AWL-Quellen. 11-3
Prozeßalarm ... 4-29
Uhrzeitalarm ... 4-25
Verzögerungsalarm ... 4-27
Weckalarm ... 4-27
zum Exportieren von Symboltabellen ... 7-15
zum Festlegen von Bausteineigenschaften in AWL-Quellen ... 11-5
zum Festlegen von Systemattributen in AWL-Quellen ... 11-4
zum Importieren von Symboltabellen ... 7-15
zur Eingabe von Anweisungen in AWL-Quellen ... 11-2
zur Reihenfolge der Bausteine in AWL-Quellen ... 11-4
Regeln für Autorisierung ... 2-5
Registerdialog ... 5-19
Remanente Speicherbereiche in S7-300-CPUs ... A-26
Remanente Speicherbereiche in S7-400-CPUs ... A-27
Remanenter Speicher
bei S7-300-CPUs ... A-26
bei S7-400-CPUs ... A-27
Remanenz
nach Netzausfall ... A-5
Reorganisieren von Projekten und Bibliotheken ... 25-2
Restzyklus ... A-6, A-9, A-11
Ringpuffer (Diagnosepuffer) ... A-24
RPL_VAL ... 21-29
Rücksetzen
Datenwerte auf die Anfangswerte ... 10-8
RUN ... A-12
Betriebszustand der CPU ... A-1
CPU-Tätigkeiten ... A-5
Runtime Software ... 1-17
S5 TIME
Bereich ... A-30
Format, Zeitbasis, Datentyp ... A-38
S7/M7-Programm ohne Station und CPU ... 5-16
S7-CFC ... 8-9
S7-GRAPH ... 8-6, 8-7
S7-HiGraph ... 8-7, 8-8
S7-Programm einfügen ... 6-6
S7-Quellen bearbeiten ... 11-13
S7-SCL ... 8-5
Sachschaden verhüten ... 18-18
SCAN-Meldung
siehe Symbolbezogene Meldung ... 14-14
Schachtelungstiefe ... 4-9
Schaltbücher drucken ... 22-1
Schnellansicht der Diagnoseinformation ... 21-5
Schnelles Positionieren auf Verwendungsstellen im Programm ... 12-12
Schritte
zum Anzeigen und Ändern des Betriebszustands ... 16-4
zum Anzeigen und Einstellen von Uhrzeit und Datum ... 16-5
zum Komprimieren des Speicherinhalts einer S7-CPU ... 17-21
SCL ... 8-3
Seitenformat
einstellen ... 22-3, 22-4
Senden
eigener Diagnosemeldungen ... 21-20
SET_CLK ... 4-26, A-117
SET_RTM ... A-117
SET_TINT ... 4-25, 4-26
Setup
Flash-File-System ... 2-11
Ident-Nummer eintragen ... 2-9
Memory Card-Parametrierung ... 2-11
SFB ... 4-23, A-39
SFB 20 STOP ... 4-11
SFB 33 ... 14-7
SFB 34 ... 14-7
SFB 35 ... 14-6, 14-7
SFB 36 ... 14-6, 14-7
SFB 37 ... 14-7
SFC ... 4-23
einsetzen ... A-18
SFC 0 SET_CLK ... 4-25, A-117
SFC 1 READ_CLK ... A-117
SFC 2 SET_RTM ... A-117
SFC 3 CTRL_RTM ... A-117
SFC 4 READ_RTM ... A-117
SFC 13 DPNRM_DG ... A-114
SFC 14 DPRD_DAT ... A-114
SFC 15 DPWR_DAT ... A-114
SFC 17/18 ... 14-6
SFC 20 BLKMOV ... A-14
SFC 22 CREAT_DB ... A-14
SFC 26 UPDAT_PI ... 4-11, A-18
SFC 27 UPDAT_PO ... 4-15, A-18
SFC 28 SET_TINT ... 4-25
SFC 29 CAN_TINT ... 4-25
SFC 30 ACT_TINT ... 4-25
SFC 31 QRY_TINT ... 4-25
SFC 32 SRT_DINT ... 4-27
SFC 36 MSK_FLT ... 4-33
Beispiel in AWL ... A-104
Beispiel in KOP ... A-104
SFC 37 DMSK_FLT ... 4-33

Beispiel in AWL ... A-104
Beispiel in KOP ... A-104
SFC 38 READ_ERR
Beispiel in AWL ... A-104
Beispiel in KOP ... A-104
SFC 39 DIS_IRT ... 4-33
Beispiel in AWL ... A-108
SFC 40 EN_IRT ... 4-33
Beispiel in AWL ... A-108
SFC 41 DIS_AIRT ... 4-33
Beispiel in AWL ... A-109
SFC 42 EN_AIRT ... 4-33
Beispiel in AWL ... A-109
SFC 44 RPL_VAL ... 21-29
SFC 46 STP ... 4-12
SFC 48 SNC_RTCB ... A-117
SFC 51 RDSYSST ... 21-17, 21-18, A-24
SFC 52 WR_USMSG ... 21-20
SFC 55 WR_PARM ... A-111, A-115
SFC 56 WR_DPARM ... A-111, A-115
SFC 57 PARM_MOD ... A-111, A-115
Sicherheitsanforderungen
Beschreibung für das Beispiel eines industriellen Mischprozesses ... 3-7
Sicherheitshinweise
Überschreiten des L-Stack ... A-21
Sicherheitsvorkehrungen zum Forcen von Variablen ... 18-20
Signalbaugruppe simulieren ... 20-1
SIMATIC Manager ... 5-1, 5-2
Baustein-Längen anzeigen ... 8-14
SIMATIC PC-Station ... 1-10
SIMATIC-Komponenten für die Meldungsprojektierung ... 14-4
Simulation einer CPU- oder Signal-Baugruppe ... 20-1
Simulationsprogramm ... 20-1
Sitzungsgedächtnis ... 5-28
Slaves mit fehlenden oder fehlerhaften GSD-Dateien ... A-73
SNC_RTCB ... A-117
Soll-Ist-Baugruppenüberwachung
Anlauf-OBs ... 4-30
Soll-Ist-Vergleich ... A-5
Sortieren
in der Querverweisliste ... 12-3
Symbole ... 7-12
Spannungslos
Betriebszustand der CPU ... A-1
Speicher ... A-28
konfigurierbar ... A-28
Speicherbereiche ... A-13
Adreßbereiche ... A-18
Arbeitsspeicher ... A-13
Besonderheiten bei S7-300 ... A-14
Besonderheiten bei S7-400 ... A-14
Ladespeicher ... A-13
remanenter Speicher ... A-26
Systemspeicher ... A-13
Speichern
Anwendungsfälle ... 22-5
AWL-Quellen ... 11-15
Bausteine ... 9-24
Datenbausteine ... 10-9
Fensteranordnung ... 5-29
Variablentabelle ... 18-3
von Codebausteinen ... 9-24
Sperren von Alarm- und Asynchronereignissen
Beispiel ... A-108
Sprache für Anzeige ... 14-16
Spracheditor starten ... 8-2
Spulen plazieren ... 9-16
SRT_DINT ... 4-27
Stack-Inhalte im Betriebszustand STOP ... 21-14
Standardbibliothek ... 6-6
Standardbibliotheken
Übersicht ... 8-19
Starten
Prozeßalarm ... 4-29
STEP 7 mit vorgegebenen Startparametern ... 5-2
STEP 7-Installation ... 2-10
STEP7 ... 5-1
Uhrzeitalarm ... 4-25
Verzögerungsalarm ... 4-27
Weckalarm ... 4-28
Startereignisse
Anlauf-OBs ... 4-30
maskieren ... 4-35
verzögern ... 4-35
Station ... 5-8
einfügen ... 6-4, 6-5
laden in PG ... 17-14
Stationskonfiguration in ein Zielsystem laden ... 17-7
Stationskonfiguration laden in das PG ... 17-16
Statuszeile
Beispiel ... 5-19
STEP 7
aufrufen der Software ... 5-1
Benutzungsoberfläche ... 5-19
deinstallieren ... 2-14
Fehler während der Installation ... 2-11
Fehler-OBs
Reagieren auf Fehler ... 4-33
Installation ... 2-8

STEP7
Basissoftware 1-6
Programmiersprachen..... 1-6
Steuern
Prinzipielle Vorgehensweise..... 18-2
Steuern von Variablen
Einführung.....18-15
Trigger festlegen.....18-16
Steuerwerte
Beispiele für die Eingabe..... 18-9
Steuerwerte einfügen..... 18-5
STOP.....A-4
Betriebszustand der CPU.....A-1
STOP-Ursache ermitteln.....21-14
Störungen eingrenzen..... 21-1
Störungsbehandlung.....21-22
STRING A-39
Stromfluß 9-18
Stromversorgungsfehler (OB 81).....21-34
Stromversorgungsfehler-OB.....21-34
STRUCT A-39
Struktur des Anwenderprogramms
"Uhrzeitalarme".....A-92
erstellen.....A-46
Ladespeicher.....A-14, A-15
von anwenderdefinierten Datentypen in AWL-Quellen..... 11-9
von Bausteinen in AWL-Quellen 11-8
von Codebausteinen in AWL-Quellen 11-8
von Datenbausteinen in AWL-Quellen 11-9
Struktur des Anwenderprogramms
"Verzögerungsalarme" A-99
Struktur und Bestandteile der Symboltabelle. 7-6
Struktur-Datentyp (STRUCT)
Anzahl verschachtelter Ebenen A-41
Beschreibung.....A-45
Strukturiert programmieren..... 4-8
Strukturierte Datentypen A-39, A-41
Feld.....A-42, A-43, A-44
Verschachtelungsstrukturen und Felder..A-41
Struktur
Verschachtelungsstrukturen und Felder..A-41
Strukturierter Datentypen
Struktur.....A-45
Strukturiertes Programm
Entwerfen.....A-78
Vorteile..... 4-2
Suchfunktion für Fehler im Anweisungsteil.. 9-14
Symbolbezogene Meldungen
zulässige Signale.....14-14
Zuordnung zur Symboltabelle.....14-14
Symbolbezogene Meldungen zuordnen
und bearbeiten.....14-14
Symbole..... 7-13, 7-14, 7-15
bei der Programmeingabe definieren.....7-11
filtern.....7-12, 7-13
Funktionsleiste 5-19
global 7-3
Groß-/Kleinschreibung..... 7-13, 7-14, 7-15
in der Programmstruktur 12-4
in eine Variablentabelle einfügen.....18-4
lokal 7-3
nicht verwendete.....12-8
sortieren.....7-12
Symbole eingeben.....7-12
Symbole für Objekte im SIMATIC Manager ...5-5
Symbolik 7-4
Symbolinformation.....7-1
Symbolische Adressierung 7-4
im Beispielprogramm A-79
Symbolischen Namen zuordnen..... A-79
Symboltabelle 7-4
BuB-Attribute projektieren 15-3
Dateiformate für den Import/Export.....7-16
für globale Symbole.....7-6
importieren/exportieren 7-15
öffnen.....7-12, 7-13
Struktur und Bestandteile.....7-6
zulässige Adressen.....7-8
zulässige Datentypen.....7-8
Synchrone Fehler
Verwenden von OBs.....4-33
Synchronfehlerereignisse
maskieren und demaskieren A-104
Synchronisieren
Uhr.....A-117
Syntax für Bausteine in AWL-Quellen.....11-10
Systemarchitektur
Betriebszustände der CPU..... A-1
Zyklus 4-11
Systemattribute
für die Meldungsprojektierung 14-7
für die PCS7-Meldungsprojektierung 14-13
für Parameter 9-5
in der Symboltabelle 7-7
Systemdaten.....21-19
Systemdiagnose
erweitern 21-20
Systemfehler.....21-21
Systemfehler melden.....14-23, 14-24, 14-27, ... 14-28, 14-29
Einstellungen.....14-26
erzeugte Fehler-OBs 14-28
erzeugter FB
DB 14-29
Systemfehler melden

unterstützte Komponenten.....14-24
Systemfunktionen 4-3, 4-23
Arten 4-23
Systemfunktionsbausteine 4-3, 4-23
Arten 4-23
Systemfunktionsbausteine (SFB) und Systemfunktionen (SFC) 4-23
Systemparameter A-114
Systemspeicher A-13
Systemzustandsliste 21-18, 21-19, 21-20
Inhalt 21-18
lesen 21-19
Systemzustandsliste SZL 21-18
SZL 21-18, 21-19
T-Abzweig 9-20
Taktmerker A-118
Tastenkombinationen
für das Bewegen des Cursors..... 5-32
für das Markieren von Texten 5-33
für das Umschalten zwischen Fenstertypen..... 5-34
für den Zugriff auf Online-Hilfe..... 5-33
für Menübefehle 5-31
Taufen von Netzteilnehmern 17-8
Tauschen von Baugruppen 25-1
Teilnehmeradresse ändern 17-8
Teilprozeßabbild A-19, A-20
aktualisieren mit SFCs A-19
systemseitig aktualisieren..... A-20
Testen
mit der Variablentabelle..... 18-1
mit Programmstatus 19-1
mit Simulationsprogramm (Optionspaket) 20-1
Testen im Einzelschrittmodus.....19-5, 19-6
Testen mit der Variablentabelle 25-3
Textbibliotheken 14-17
übersetzen.....14-17, 14-18
Texte mehrsprachig verwalten..... 6-9
Textlisten
siehe Listen bedienerrelevanter Texte....14-16
TIMER.....A-49, A-50
Parametertyp A-49
Tips und Tricks 1-10, 25-1, 25-2, 25-3, 25-5
Titel
zu Bausteinen..... 9-13
zu Netzwerken 9-13
Transferieren der Projektierungsdaten zum BuB-Zielsystem..... 15-5
Trigger zum Beobachten von Variablen festlegen.....18-13
Trigger zum Steuern von Variablen festlegen.....18-16
Triggerbedingung 18-13
Triggerbedingungen für die Aufzeichnung des Programmstatus.....19-9
Triggerpunkte
einstellen.....18-13
TTR Typ.....1-10
Typdatei..... A-72
Übergabe an IN_OUT-Parameter eines FB A-69
Übergänge
von Betriebszuständen A-1
Übergeben von Parametern
Beispiel-FB für das Beispiel eines industriellen Mischprozesses..... A-81
Entwerfen von Parametern für ein strukturiertes Programm..... A-81
Parametertypen..... A-49
Speichern der übergebenen Werte.....4-17
Übermittlung von Diagnoseinformationen ..21-17
Überschreibmodus 9-14
Überschreiten des L-Stack..... A-21
Übersetzen
AWL-Quellen.....11-16
Übersetzen und Bearbeiten
von bedienerrelevanten Texten 14-16
Übersicht
der möglichen Referenzdaten.....12-1
Meldebausteine 14-6
Standardbibliotheken 8-19
Überwachungszeiten 4-32
UDT 8-11, A-39
Korrigieren der Schnittstelle 9-24
UDTs
Struktur eingeben 10-6
UDTs in AWL-Quellen
Beispiel 11-24
Uhr
parametrieren A-117
synchronisieren A-117
Uhrzeit
ändern.....4-26
lesen A-117
stellen A-117
Uhrzeit (TIME OF DAY)
Bereich..... A-30
Uhrzeitalarm
abfragen.....4-25
deaktivieren.....4-26
Priorität 4-26
Regeln 4-25
starten.....4-25
Uhrzeit ändern.....4-25
Uhrzeitalarme..... 4-25, 4-26
Struktur A-92
Uhrzeitformat A-117

Uhrzeitfunktionen A-117
Umbenennen
Projekte 5-21, 5-22
Umschalten zwischen Fenstertypen 5-34
Umsetzen A-72
Version 1-Projekte A-70
Version 2-Projekte A-71
Umverdrahten
Bausteine 8-15
Operanden 8-15
Ungepufferter Betrieb
remanente Speicherbereiche A-27
Unterbrechungs-Stack A-13, A-22
Unterbrechungszeit A-5
Untergliedern des Prozesses am Beispiel
eines industriellen Mischprozesses 3-2
Unterschied zwischen Speichern und Laden
von Bausteinen 17-2
Unterschiede zwischen Forcen und Steuern
von Variablen 18-21
Unterstützte Komponenten und
Funktionsumfang 14-24
Unvollständige und mehrdeutige Symbole
in der Symboltabelle 7-9
Unzulässige Verschaltungen in KOP 9-18
UPDAT_PI 4-15, A-18
UPDAT_PO 4-11, A-18
Upload (Netzkonfiguration laden in PG) 17-17
Urlöschen A-4
der CPU 17-19
U-Stack
Beschreibung A-22
Verwendung durch den Systemspeicher . A-22
Variablen
bedienen und beobachten 15-1
beobachten 18-13
steuern 18-15
Variablendeklaration in AWL-Quellen
Beispiele 11-18
Variablendeklarationstabelle 9-1, 9-5, 9-6
Aufgabe 9-5
FB für das Beispiel eines industriellen
Mischprozesses A-81
FC für das Beispiel eines industriellen
Mischprozesses A-85
für OB81 21-24
Multiinstanz eingeben 9-10
OB für das Beispiel eines industriellen
Mischprozesses A-87
Struktur 9-6
Systemattribute für Parameter 9-5
Variablentabelle 18-3
bearbeiten 18-4
Beispiel 18-4
Beispiel für die Eingabe von Operanden .. 18-8
kopieren/verschieben 18-3
Maximale Größe einer 18-4
Nutzen 18-1
Operanden oder Symbole einfügen 18-4
speichern 18-1, 18-3
Syntaxprüfung 18-5
Ventile
Eingangs-/Ausgangs-Diagramm erstellen .. 3-7
Verbindung
zur CPU herstellen 18-12
Verbindungen
in das PG laden 17-17
Verbindungs-Ressourcen 1-10
Verbindungstabelle 1-10
Vergabe von Meldenummern 14-5
Verhältnis zwischen
Betriebszuständen der CPU A-1
Verlust der Autorisierung 2-1
Vermeiden von Fehlern beim Aufrufen von
Bausteinen 9-25
Verschachtelte Aufrufe von Codebausteinen
Auswirkungen auf den B-Stack und den
L-Stack A-22
Verschieben
Objekt 5-21, 5-22, 5-23, 5-24, 5-26
Version 1-Projekte umsetzen A-70
Version 2-Projekte umsetzen A-71
Verwenden
anwenderdefinierten Datentypen für den
Datenzugriff A-47
Felder für den Datenzugriff A-42
Parametertyp ANY A-58
Parametertyp POINTER A-52
Strukturen für den Datenzugriff A-45
Systemspeicherbereiche A-16
Taktmerkern und Zeiten A-118
von älteren Projekten A-70, A-71
zusammengesetzte Datentypen A-41
Verwendung der Variablendeklaration in
Codebausteinen 9-5
Verwendung von Multiinstanzen 9-9
Verzögern
Startereignisse 4-35
Verzögerte Bearbeitung von Alarm- und
Asynchronereignisse
Beispiel A-109
Verzögerungsalarm
Priorität 4-27
Regeln 4-27
starten 4-27
Struktur A-99

Virtuellen Arbeitsspeicher einstellen 25-5
Voraussetzungen
für das Archivieren 22-6
für das Laden..... 17-1
Voreinstellungen für Programmeditor
KOP/FUP/AWL 9-2
Vorgefertigte Bausteine..... 4-23
Vorgehen beim Installieren..... 2-9
Vorgehensweise
beim Archivieren/Dearchivieren 22-6
beim Eingeben von Anweisungen..... 9-12
für M7-Systeme 24-1
Warmstart A-5
Warnung
Überschreiten des L-Stack A-21
Was ist neu in STEP 7 Version 5.1..... 1-10
Wechsel der Betriebszustände..... A-1
Weckalarm
Regeln..... 4-27
starten 4-28
Weckalarme 4-27, 4-28
Wegweiser 1-1
Welche Meldebausteine gibt es?..... 14-6
Welche Meldeverfahren gibt es? 14-1
Wiederanlauf A-2, A-5, A-6, A-7, A-8, A-9
abbrechen A-5
automatisch A-6
manuell..... A-5
Wiederherstellen
Autorisierung 2-1
Fensteranordnung Fensteranordnung
speichern..... 5-29
Wissenswertes
zum Betriebszustand "HALT"..... 19-7
zum Testen im Einzelschrittmodus /
Haltepunkte..... 19-5
WORD
Datentyp..... A-37
Wort (WORD)
Bereich..... A-30
Datentyp..... A-29
WR_DPARM A-112, A-116
WR_PARM..... A-112, A-116
WR_USMSG 21-20
Zahlendarstellung
binär-codierte Dezimalzahlen (BCD)..... A-37
Bit A-29
Byte..... A-29
Datum und Uhrzeit (DATE AND TIME) ... A-40
Doppelwort A-29
Ganzzahl (16 Bit) A-31
Ganzzahl (32 Bit) A-32
Gleitpunktzahlen A-33, A-34, A-35
S5 TIME..... A-38
Wort..... A-29
Zähler
Belegungsplan.....12-7
Obergrenzen für die Eingabe 18-7
Speicherbereich
remanent..... A-26
Zeichen (CHAR)
Bereich..... A-30
Zeitbasis für S5 TIME A-38
Zeiten
Belegungsplan.....12-7
Obergrenzen für die Eingabe 18-6
Zeiten (T) A-118
Speicherbereich
remanent..... A-26
Zeitfehler (OB 80).....21-33
Zeitfehler-OB.....21-33
Zeitstempel
als Bausteineigenschaft 13-3
bei globalen Datenbausteinen 13-5
bei Instanz-Datenbausteinen.....13-6
bei UDTs und von UDTs abgeleiteten
DBs.....13-7
in Codebausteinen.....13-4
Zeitstempelkonflikte.....13-3
Zerlegen des Prozesses in Aufgaben und
Bereiche.....3-2
Zerlegen eines Prozesses in Aufgabenbereiche
für das Beispiel eines industriellen
Mischprozesses.....3-2
Ziehen-/Stecken-Alarm (OB 83) 21-36
Ziehen-/Stecken-Alarm-OB.....21-36
Zielsystem
Bausteine nachladen 17-5
Zugangsberechtigung.....16-3
Zugriff auf den Peripheriedatenbereich.....A-111
Zugriff auf den Prozeßdatenbereich A-110
Zugriffsrechte auf Bausteine bzw. Quellen.....9-3
Zulässige Adressen und Datentypen in der
Symboltabelle.....7-8
Zulässige Bausteineigenschaften
je Bausteinart 11-7
Zulässige Datentypen beim Übergeben
von Parametern..... A-64
Zuordnen
symbolischen Namen..... A-79
Zuordnen von Datentypen zu Lokaldaten
von Codebausteinen..... A-62
Zuordnen von Speicher
im L-Stack..... A-21, A-22
Zurückladen
Bausteine aus S7-CPU 17-15

Zurückladen einer Konfiguration aus einer Station 17-16
Zurückladen einer Netzkonfiguration (Laden in PG) 17-17
Zusammengesetzte Datentypen A-39
Zusammenspiel zwischen Variablendeklarationstabelle und Anweisungsteil 9-6
Zustandsgraph 8-7
Zyklische Programmbearbeitung..... 4-4, 4-7, 4-8
Zyklus 4-4, 4-5, 4-11, 4-12, 4-13
Zyklusbelastung durch Kommunikation 4-15
Zyklusüberwachungszeit.....4-13
Zykluszeit.....4-12, 4-13, 4-15, 4-16
Zykluszeit zur Vermeidung von Zeitfehlern kontrollieren.....21-16

An
Siemens AG
A&D AS E 81
Östliche Rheinbrückenstr. 50
76181 Karlsruhe

Absender:

Ihr Name: .....

Ihre Funktion: .....

Ihre Firma:.....

Straße: .....

Ort: .....

Telefon: .....

Bitte kreuzen Sie Ihren zutreffenden Industriezweig an:

- ❒ Automobilindustrie
- ❒ Chemische Industrie
- ❒ Elektroindustrie
- ❒ Nahrungsmittel
- ❒ Leittechnik
- ❒ Maschinenbau
- ❒ Petrochemie
- ❒ Pharmazeutische Industrie
- ❒ Kunststoffverarbeitung
- ❒ Papierindustrie
- ❒ Textilindustrie
- ❒ Transportwesen
- ❒ Andere .....

Vorschläge und Anmerkungen zur Anwenderdokumentation

Ihre Anmerkungen und Vorschläge helfen uns, die Qualität und Benutzbarkeit unserer Dokumentation zu verbessern. Bitte füllen Sie diesen Fragebogen bei der nächsten Gelegenheit aus und senden Sie ihn an Siemens zurück.

Geben Sie bitte bei den folgenden Fragen Ihre persönliche Bewertung mit Werten von 1 = gut bis 5 = schlecht an.

1. Entspricht der Inhalt Ihren Anforderungen? ☐
2. Sind die benötigten Informationen leicht zu finden? ☐
3. Sind die Texte leicht verständlich? ☐
4. Entspricht der Grad der technischen Einzelheiten Ihren Anforderungen? ☐
5. Wie bewerten Sie die Qualität der Abbildungen und Tabellen? ☐

Falls Sie auf konkrete Probleme gestoßen sind, erläutern Sie diese bitte in den folgenden Zeilen:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....